entropy

MDPI

Article

Multi-Agent System Supporting Automated Large-Scale Photometric Computations

Adam Sędziwy * and Leszek Kotulski

Department of Applied Computer Science, AGH University of Science and Technology,
al. Mickiewicza 30, 30-059 Kraków, Poland; kotulski@agh.edu.pl

* Correspondence: sedziwy@agh.edu.pl; Tel.: +48-12-617-5204

Academic Editor: J. A. Tenreiro Machado
Received: 2 February 2016; Accepted: 23 February 2016; Published: 27 February 2016

Abstract: The technologies related to green energy, smart cities and similar areas being dynamically developed in recent years, face frequently problems of a computational nature rather than a technological one. The example is the ability of accurately predicting the weather conditions for PV farms or wind turbines. Another group of issues is related to the complexity of the computations required to obtain an optimal setup of a solution being designed. In this article, we present the case representing the latter group of problems, namely designing large-scale power-saving lighting installations. The term "large-scale" refers to an entire city area, containing tens of thousands of luminaires. Although a simple power reduction for a single street, giving limited savings, is relatively easy, it becomes infeasible for tasks covering thousands of luminaires described by precise coordinates (instead of simplified layouts). To overcome this critical issue, we propose introducing a formal representation of a computing problem and applying a multi-agent system to perform design-related computations in parallel. The important measure introduced in the article indicating optimization progress is entropy. It also allows for terminating optimization when the solution is satisfying. The article contains the results of real-life calculations being made with the help of the presented approach.

Keywords: street lighting; energy efficiency; photometric computations; entropy; multi-agent system; graphs

1. Introduction

The rapid development of technologies and their practical applications opened a series of problems that have to be addressed. The example is the domain of energy management and distribution. Besides the strictly technological issues, like those related to the chemistry of energy storage units or the durability of materials, for instance, there is still many more abstract problems, e.g., concerning the computability or environment state (in the broad sense) forecasting. In this article, we focus on the problem of the former type, which arose when LED sources began to be widely used in public lighting. It may be stated as follows: how does one configure public lighting installations in a given urban area, and how does one control them, so that the achieved power (money) savings are the highest? This question is timely, because according to the report published by Northeast Group [1], the number of streetlights installed globally is estimated to increase by 60 million and reach nearly 340 million by 2025, which means that the expected annual energy costs will reach $23.9 billion to $42.5 billion by 2025, dependent on the assumed operational power.

Although the question formulated above looks like an optimization problem, it covers, however, two issues: how does one design (retrofit) energy-efficient street lighting in the scale of an entire urban area, and what is the control strategy (for them) leading to the maximum power savings? In this article, we present the solution for both sub-problems that was developed by our team.

Since reducing the power usage of a single luminaire by one or two percent only is a reasonable result, due to the effect of scale, the range of research focusing on roadway lighting energy efficiency were made. They may be divided into three groups:

1. optimal design improvement [2,3],
2. control methods [4],
3. hardware-based approaches, e.g., introducing fixtures with higher "lumens per watt" ratios or improving the reflective properties of a road surface by using some chemical additions [5].

The considerations presented in this paper fall into the first area, although it will be shown that they can be successfully applied to the control tasks, as well. The notion of *an optimization of a lighting design* being used in the article refers to a process of searching for the values of the parameters, such as the fixture model, arm length, pole height, fixture's tilt and dimming, that yield the minimum power usage of an installation. Since the consumed power decreases with increasing dimming level, the last property is crucial for the quality of the final result.

There exists a design approach introduced in [6] yielding the significant improvement of the energy efficiency, but it generates the rapid growth of the required computations, which cannot be handled by a human designer in a reasonable time. To overcome this issue, a computing problem has to be properly formalized (to enable computer-aided processing), and a multi-agent system performing parallel computations has to be involved. The role of agents in the considered context will be discussed in Section 5. To track the results of the agents' work and to terminate the optimization process, we will use entropy, introduced in Section 3.

A practical application of computational methods of large-scale designs shows that the computing time for a centralized problem is still unacceptable in practice. For that reason, it is necessary to decompose it into multiple sub-problems to be solved in parallel. In this article, we propose using a graph-based, formal representation of a lighting design problem, based on a hierarchical hypergraph model of physical objects [7]. This model can be easily distributed and processed in parallel. Moreover, we propose using a multi-agent system to perform those parallel computations: such an approach combines a graph formalism with agent system methods, and thus, it allows for automated, scalable and time-efficient photometric calculations required in large street lighting optimizations made prior to retrofit works, for example when old sodium lamps are to be replaced with new, LED ones. This approach can be applied to a customized design and make this method applicable to practical use.

The article is arranged as follows. Preliminaries concerning the lighting design process are presented in Section 2. In Section 3, we introduce the notion of entropy, being crucial for the lighting design quality assessment and the computation progress control. In Section 4, the graph representation underlying the automated lighting design computations is defined. It also contains a decomposition method for distributing graph-based models. A multi-agent system perspective of computations is shown in Section 5. The obtained results and final conclusions can be found in Sections 6 and 7, respectively.

2. State of the Art and Preliminaries

As it has been mentioned in the previous section that a design method [6] was proposed producing lighting installations having the minimal power consumption. This method, however, can be applied for small situations only due to the large computational overhead: each situation is computed luminaire by luminaire. This makes this task infeasible for a human. For that reason, all industry-standard computer programs use some simplifications, rather than exact data, to facilitate a design process. All of those issues will be discussed in depth in Section 2.1.

To meet the power efficiency demands and to overcome the complexity issue, a graph-based formalization of a problem has to be applied together with a parallel processing performed by a multi-agent system. In Section 2.2, the structure of such a calculation process will be sketched.

Graph and hypergraph models have been broadly used for many years in a range of problems in multiple domains, such as chemistry [8], biology [9,10] or computer-aided design [11]. They are applied as both centralized and distributed models [12]. This is due the high expressiveness of (hyper-)graph formalisms, which reflect the properties of both natural (physical) and abstract (logical) systems. A role graph and hypergraph structures modeling an urban space and a relevant lighting infrastructure are crucial in the considered context. Thanks to these, any computational task related to an infrastructure optimization can be accomplished with the help of a computer system, which requires, in turn, well-defined and structured input data and a problem description.

2.1. Lighting Design Overview

In the commonly-used approach to street lighting design, two practices can be distinguished: The overall design process is viewed as a compound of multiple, separate processes focusing on *simple lighting situations*. The example of such a *simple lighting situation* is the single road section, road junction or a section consisting of the walkway, carriageway and bike lane. The second practice is that the road and luminaire layouts are simplified, *i.e.*, a road is assumed to have a constant width with no irregularities with identical luminaires distributed along it evenly. Such an assumption is for practical reasons: the time required for the preparation a design for non-uniform layouts is significantly higher than for regular ones. It may be even worse if the energy efficiency optimization is performed for an irregular layout. Usually, creating a lighting design is a kind of trial-and-error procedure. A human designer "manually" prepares a project and then verifies it against some criteria. In the considered context, it is the energy efficiency of an installation. The above verification is supported by market-available software [13]. If a project check fails, then a new (modified) configuration has to be examined.

It was proven in [6] that the street lighting power usage may be reduced up to 15% when using accurate layout data instead of the simplified layouts. Such accurate computations are based on raw geolocalized coordinates of luminaires and streets, which may be obtained either from existing digitalized inventory data (e.g., CAD files) or acquired in the field. Determining a calculation field (carriageway, crossroad, and so on) on the basis of the raw GIS data can be done manually for a single area. For larger spaces, however, this becomes a time-consuming process, which is not doable by a human. It should be noted that besides the calculation field identification, a set of relevant luminaires ascribed to it has to be determined, as well. From the computational perspective, the process can be described as arranging the raw input data points into well-defined structures corresponding to streets and luminaires.

To overcome the computational complexity problems, the following conditions have to be satisfied.

- A human has to be excluded from the design process. This is required because human activity is the bottleneck of the process.
- To enable the computer-aided, fully-automated data processing and design preparation, some formal representation of a problem has to be available.

The roadway lighting is regulated by multiple standards, recommendations and good practices. The set of key standards for European countries is contained in the bundle EN 13201, which, among others, defines lighting classes for public spaces, performance requirements for them and introduces the methodology of computing and measuring photometric parameters. Two standards are particularly important in this article: the performance requirements (EN 13201-2) and the calculation of performance (EN 13201-3) [14,15]. The EN 13201-2 specifies what are the mandatory performance values of selected photometric parameters. For example, Table 1 presents the requirements of so-called ME-lighting classes which establish performance requirements for traffic routes of medium to high driving speeds. It introduces the minimum acceptable values for the light reflected on the road surface (*i.e.*, luminance), luminance uniformity and others.

Table 1. ME-lighting classes according to EN 13201-2.

Class	L_{avg} (cd/m^2) (min. *)	U_o (min.)	U_l (min.)	TI (%) (max. **)	SR (min.)
ME1	2.0	0.4	0.7	10	0.5
ME2	1.5	0.4	0.7	10	0.5
ME3a	1.0	0.4	0.7	15	0.5
ME3b	1.0	0.4	0.6	15	0.5
ME3c	1.0	0.4	0.5	15	0.5
ME4a	0.75	0.4	0.6	15	0.5
ME4b	0.75	0.4	0.5	15	0.5
ME5	0.5	0.35	0.4	15	0.5
ME6	0.3	0.35	0.4	15	n/a

* min.: minimum allowed value, ** max.: maximum allowed value.

EN 13201-3 gives the detailed receipt for computing all photometric parameters. This receipt is used by all market-available tools, like DIALux, Relux, Agi32, and others (see [13]), which support photometric computations. All of them verify if a project complies with the constraints given in Table 1 (in the case of roadways), assuming the uniformity of a lighting situation: evenly-spaced and identical luminaires, uniformly distant from a carriageway having a constant width. Calculations are made for a road section (also known as a calculation field) delimited by two subsequent luminaires (Figure 1b), which is representative of the entire road due to its assumed symmetry. Further calculating details, which can be found in [14], will not be discussed here.

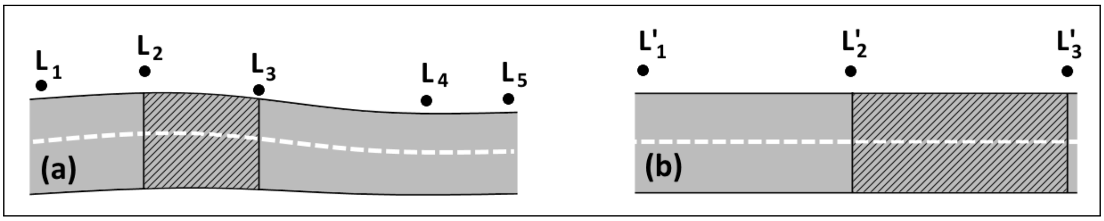


Figure 1. (**a**) Actual road layout with physical luminaires $L_1, L_2 \ldots$; and (**b**) its simplified form. $L'_1, L'_2, \ldots$ denote "averaged" luminaire positions. In both cases, the sample calculation fields are filled with an oblique hatching.

When performing computations for real-life cases, the most pessimistic conditions are assumed: the road width and lamp spacing are set to maxima or averages of actual widths and spacings, respectively (see Figure 1). Thus, the obtained results can be expected to be standard compliant (in fact, aggregating with the average may lead to violating the standard, as shown in [6]). The side effect of this approach is energy wasting and light pollution caused by over-illumination.

The proposed approach to improving energy efficiency applies the method discussed in depth in [6]. It uses the customized design based on actual GIS data, applied in the place of the standard methods described above. For each point of a calculation field (Figure 1a), the relevant luminaires have to be identified (Figure 2) prior to photometric computations. According to the standard [14], this is accomplished by selecting all luminaires falling into the region of the size $17H \times 10H$, containing a calculation point, where H is a pole height. Note that such an identification requires processing raw GIS data and presumably some additional information concerning, for example, a luminaire type (park luminaires may be ignored when computing roadway lighting). Moreover, some luminaires falling into such a region may be already processed. Then, an actual calculation must not corrupt the results obtained previously. The problems mentioned above influence the complexity of a design task. In the sequel, we present how a customized design can handle large-scale computations, covering thousands of calculation fields.

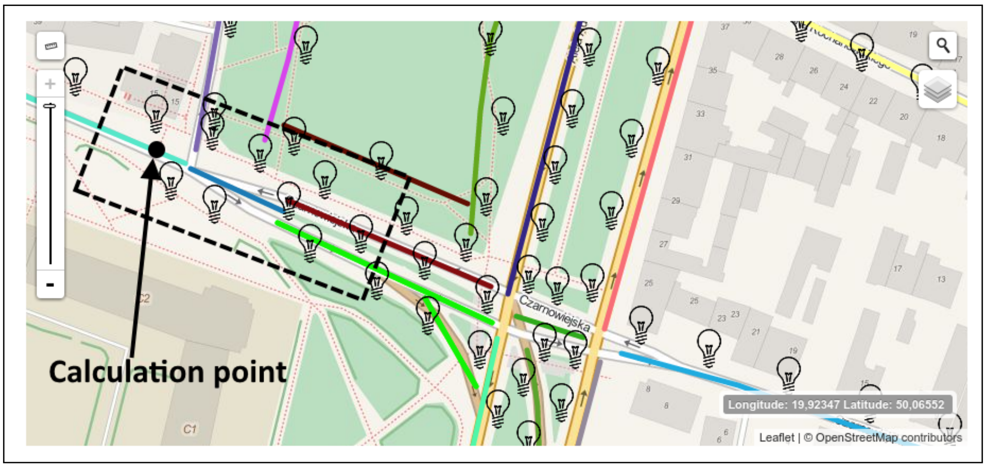

Calculation point
Czarnowiejska
Longitude: 19,92347 Latitude: 50,06552
Leaflet | © OpenStreetMap contributors

Figure 2. The real-life case of the sample calculation point. The dashed line delimits the region containing luminaires that has to be taken to compute luminance and illuminance at the point.

This schema of photometric computations can be successfully applied for lighting control purposes. In the work [16], the notion of a *lighting profile* was presented. Since a profile defines luminaire settings for specific environment conditions (rain, dusk, high traffic, and so on; also, any combinations of them), an adaptive control is accomplished by switching among profiles on the basis of incoming information from a sensor layer. The preparation of a control logic is made by the pre-calculation of all required lighting profiles for all corresponding luminaires and areas.

2.2. Calculation Process

The considered problem of a large-scale lighting design may be described as follows. A city map consists of two layers: a background layer, including street layout data, and an upper one containing the positions of luminaires and the other related information.

As introduced in Section 4, a map is represented by a graph structure, and all important street and luminaire data are stored either in edge/node attributes or directly in a graph (see Definition 1).

Each luminaire is described, among others, by the high precision GIS coordinates of a pole, in some geodetic coordinate system, e.g., UTM or WGS84. A street layout is also defined using geodetic coordinates. The task to be accomplished is two-fold: extracting calculation fields from those raw data and making photometric computations on them. The former task is based on both map layers: it requires a street geometry to be known, but also the locations of particular luminaires, which influence the calculation field placement. The latter step is an optimization, which can be described as searching for such values of luminaire adjustments that yield the most energy-efficient installation complying with a mandatory (for an underlying street) lighting standard. Both steps are made in an interleaved manner in the course of processing.

The key step in photometric computations is determining a calculation field F with respect to the considered luminaires. For roadway lighting situations, this is accomplished by selecting two subsequent luminaires, say L_1, L_2. Figure 1 shows the fields for the customized (Figure 1a) and standard (Figure 1b) computational approaches. The next step is selecting all luminaires that have to be taken into account when computing photometric parameters on F and optimal adjustments for L_2 (L_1 is assumed to be already adjusted). After that, a subsequent luminaire is taken (L_3); the new calculation field is set; and the procedure is iterated. The detailed description of the process may be found in [6].

Note that the above schema may become complicated when there are two luminaires, say L_6, L'_6, located close to each other (see Figure 3). Then, it is necessary to reject one of them on the basis of some additional information, like pole GIS coordinates, fixture type, pole height, arm orientation with respect to the road axis, *etc*. Such an analysis is made by a *computing agent* described in Section 5.

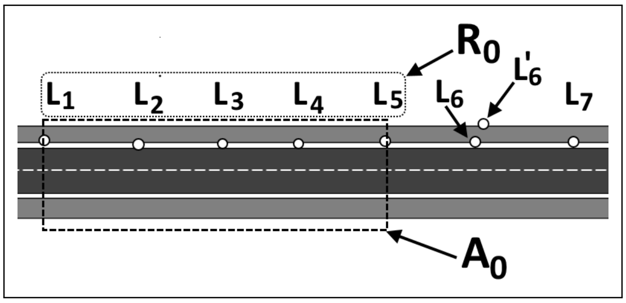
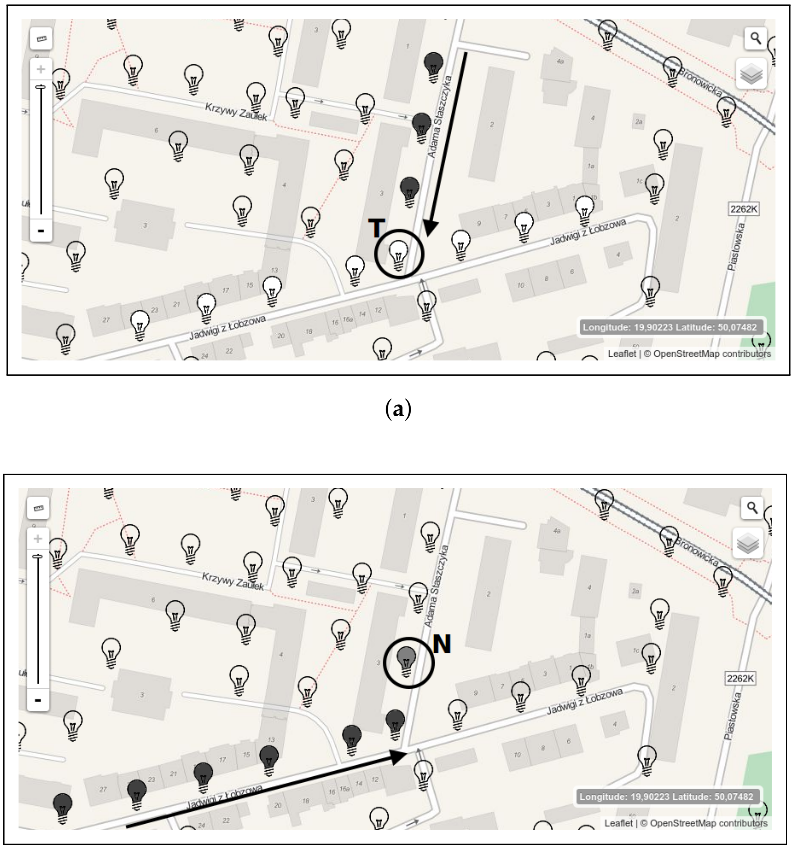


Figure 3. The initial phase of a computing process. R_0 is an initial row of luminaires, and A_0 is an area grouping all neighboring communication routes.

When considering a large set of crossing streets, two situations can occur: accessing a luminaire already processed (Figure 4a) and detecting two luminaires belonging to different series (Figure 4b).


Krzywy Zaułek
Adama Staszczyka
Jadwigi z Łobzowa
Bronowicka
Piastowska
2262K
T
N
Longitude: 19,90223 Latitude: 50,07482
Leaflet | © OpenStreetMap contributors

(**a**)

(**b**)

Figure 4. Two special cases in a calculation process. For both figures, the bold arrows indicate processing order. White bulbs denote adjusted luminaires, and dark gray ones are currently being adjusted. Non-filled bulbs stand for luminaires waiting for processing. (**a**) The calculation process meets a processed and tagged luminaire (T); (**b**) the calculation process meets a *forking point*. The N symbol denotes the first luminaire in the new segment.

The principal computing process begins with a sequence of the five subsequent luminaires grouped in a row R_0, for which photometric computations are made (Figure 3). The calculating area (not to be confused with a calculation field), A_0, being a section of a roadway and neighboring walkways (or any other communication routes) corresponding to R_0, is determined on the basis of the underlying GIS-based street layout. The results of the computations are adjustments for luminaires in R_0, such that lighting standards requirements for A_0 are met and the power usage is minimized. After completing this stage, calculations proceed iteratively: the next luminaire, l (L_6 in Figure 3), is taken, $R_1 = R_0 \cup \{l\}$, and photometric parameters are calculated for the extended area, $A_1 = A_0 \cup A(l)$, where $A(l)$ denotes an area influenced by the light emitted by l. Note that adjustments are calculated for l only, and those obtained for R_0 are not changed anymore. All luminaires being already adjusted are tagged appropriately.

When following the scenario described above, two situations can occur (see Figure 4):

1. A subsequent luminaire, say l_t, is already tagged, which means that it was proceeded with and adjusted accordingly (Figure 4a).
2. A subsequent luminaire, l_f, is located in a *forking point*, where a new segment begins, and l_f affects it (Figure 4b).

In the former case, l_t is recalculated with respect to the area being currently processed. Having two sets of settings obtained (previous and actual ones; both are assumed to comply with the relevant lighting standard requirements), one selects the more restrictive one.

Example 1. Let us assume, for example, that the carriageway of the ME5 class was processed first (the road along the dark gray symbols in Figure 4a). In the result, the power of l_t (marked with the circle and labeled with T) was adjusted to 100 W. Next, the photometric calculations for the neighboring route of the ME3a class were made (the road along the white symbols), in particular for l_t being also ascribed to this roadway. In that case, however, l_t power was adjusted to 120 W. To ensure a standard compliant solution for both streets (with ME5 and ME3a classes, respectively), one selects the latter (higher) wattage, namely 120 W.

In the second situation a new, separate process for the new segment has to be initiated for l_f. If applicable, the last processed luminaire gets shared as the first one in the new segment.

A process described in this section, being applied to the large sized problem, requires involving a calculation paradigm, which allows parallel and scalable autonomous analysis and decision making. A multi-agent system (MAS) is a suitable framework enabling that. Prior to introducing MAS, however, a formal representation of a problem, relying on the graph model (Section 4), has to be prepared for agents. It is also necessary to establish a stopping criterion for an MAS activity. This criterion, reflecting the optimization progress (the quality of a solution), will be based on the notion of entropy introduced in the next section. In Section 5, an MAS structure will be discussed.

3. Statistical Measure of a Solution Search Process

In statistical thermodynamics, the entropy of a system is defined as:

$$S = k \ln \Omega, \tag{1}$$

where k is Boltzmann's constant and Ω stands for a number of microstates yielding a given macrostate of a system. We will adopt the above formula for the considered context, *i.e.*, a multi-agent system optimizing public lighting in an urban space. The new formula will be used for two purposes: as a measure of the overall progress of a solution search process performed by an MAS and as a measure of solution quality. In order to achieve both goals, we modify Equation (1) by replacing Boltzmann's constant with an actual total power of all luminaires, $P_{tot} = \sum_i P_i$, where the sum iterates over all luminaires, and Ω with $\Omega_{conf} = N_{conf} + e$, where N_{conf} denotes the total number of possible adjustments for all luminaires taken as a whole and e is the base of the natural logarithm:

$$S_P = P_{tot} \ln \Omega_{conf} = P_{tot} \ln(N_{conf} + e). \quad (2)$$

Note that, since the P_i value for an unadjusted luminaire is a nominal power of its fixture (*i.e.*, without dimming), P_{tot} is likely to decrease in the course of an optimization.

In turn, since the value of N_{conf} is zero when all luminaires are configured, we need to add some positive number to it to stay within the domain of the logarithm. If this positive number is equal to e, then the entropy obtained for a fully-configured installation is equal to the total power of all luminaires: $S_P = P_{tot}$.

Thus, the performance of all luminaires can be regarded as a macrostate, which is intended to comply with the standard for all relevant roadways, walkways, and so on. The configurations of luminaires (being subject to changes during the optimization process) play the role of microstates. Different luminaires' settings may yield the same total power P_{tot}. Once the luminaire's optimal settings are found, it is "frozen" and excluded from the proceeding optimization, so the number of microstates, Ω_{conf}, is decreased, which means that entropy is decreased, as well. To summarize, one can regard the optimization process performed by agents as the reduction of a system's entropy. It can be accomplished in a two-fold manner: by configuring subsequent luminaires and by minimizing their power usage. The logarithmic factor in Equation (2) indicates the overall progress of the optimization. The quality of a solution can be assessed by P_{tot}.

Example 2. To assess the value of S_P, we make the following assumptions: (i) the entire lighting system consists of $N = 10{,}000$ identical luminaires; (ii) the configuration of each luminaire relies on the five degrees of freedom listed in Table 2, which means that the number of available configuration for a single luminaire is $n = 6.916 \times 10^7$. Both assumptions yield the value $\ln \Omega_{conf} = \ln(e + n^N) \approx 1.81 \times 10^5$. In turn, the order of magnitude for P_{tot} can be estimated as $P_{tot} = N \times 37.5\,\text{W} = 37.5 \times 10^4\,\text{W}$, if assuming that the average dimming level is 37.5% (note that the dimming at the level of 0% means that a fixture is not dimmed) and the average non-dimmed fixture's power is 60 watts. Thus, we obtain the approximate value of entropy:

$$S_P = 6.77 \times 10^{10}\,\text{W}.$$

Table 2. Variable parameters of a luminaire configuration. The total number of variants $n = 69{,}160{,}000$.

Parameter	Range	Step	Number of Variants
Pole height	6–12 m	0.5 m	13
Arm length	0–3 m	0.5 m	7
Fixture inclination	0°–20°	5°	5
Fixture dimming level	0%–75%	1%	76
Fixture model	n/a	n/a	2000

4. Graphs

To overcome the computing complexity issue described above, the design problem has to be formulated using some formal model enabling computer processing. Such a model should be scalable and, on the other hand, capable of hosting multi-agent system. The graph representations satisfy those demands. They are expressive enough to represent a broad range of systems [17–19]. Moreover, they were shown to be a suitable environment for multi-agent systems' deployment [20,21].

The introduced graph representation covers two levels of a system's granularity: At the lower level, one deals with a fine-grained model of physical objects, in particular buildings and surfaces (streets and other areas). An atomic portion of data corresponding to a 3D object is represented by a solid hypergraph (S-hypergraph) (see Definition 1), while any compound of such objects by an aggregated hypergraph (A-hypergraph) (Definition 2), belonging, in turn, to the upper level of the representation.

The important feature of this model is that the upper level is given by a regular graph, which may be distributed to a slashed form (Definition 3, [22]), being highly efficient from the computational point of view.

The following subsections formally introduce the above notions.

4.1. S-hypergraph, A-hypergraph

Suppose a 3D-object S consisting of faces (either planar or curved), physical edges and physical vertices (contrary to graph edges and graph vertices) is given. The object S can be represented by a hypergraph in such a way that hypergraph nodes represent object's faces, hypergraph edges correspond to physical edges and hypergraph hyperedges represent vertices of S.

This informal description may be framed as follows.

Definition 1. An *S-hypergraph* (solid hypergraph) G_S of an object S is a tuple of the form:

$$G_S = (N, A, H, lab_{\{N,A,H\}}, att_{\{N,A,H\}}),$$

where N is a nonempty set of nodes, $A \subset P_2(N)$ is a set of edges, $H \subset \bigcup_{i>2} P_i(N)$ is a finite set of hyperedges, $lab_\mu : \mu \to \mathcal{L}_\mu$ for $\mu = N, A, H$ is a labeling function for vertices, edges and hyperedges, respectively, with corresponding set of labels $\mathcal{L}_\mu$ and $att_\mu : \mu \to \mathcal{A}_\mu$ for $\mu = N, A, H$ is an attributing function for vertices, edges and hyperedges, respectively, with corresponding set of attributes $\mathcal{A}_\mu$.

When a face f of S or edge r of S is curved, then relevant geometric (analytical) data are stored in $\mathcal{A}_N(v_f)$ or $\mathcal{A}_A(e_r)$, where $v_f \in N$ denotes a hypergraph node representing a face f and $e_r \in A$ stands for a hypergraph edge representing a physical edge r.

Figure 5 shows the hypergraph corresponding to a cuboid containing faces $f_1, f_2, \dots f_6$. Since the hypergraph's structure reflects the morphology of the solid, only the detailed properties have to be stored in attributes.

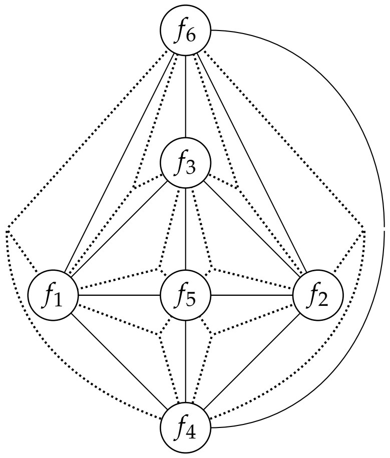


Figure 5. The solid hypergraph (S-hypergraph) representing a cuboid B_i shown in Figure 6. Note that the shape related details are hidden in the attributes of nodes, edges and hyperedges.

The above representation can be easily extended to surfaces by assuming that, to a given surface, an infinitesimal height ϵ is assigned. Such a generalization allows for the uniform representation of an urban space, including both buildings and areas. In the considered context, the nodes of an S-hypergraph represent faces of buildings and surfaces of streets, squares and other areas. Hyperedges (*i.e.*, $e \in H$) model physical vertices of 3D objects (buildings) and streets, squares, *etc.*, as well. Hypergraph edges (*i.e.*, $e \in A$) represent either physical edges of 3D objects or edges of streets, squares and other areas. The physical properties of objects will be stored as values of attributes. Note that it is also possible to represent the point objects, such as fixtures, sensors, and so on. They can be modeled with graph vertices and marked with some special attribute indicating their status.

For modeling the complex structures containing the multiple 3D-objects, we "escape" to the upper level representation based on the notion of the A-hypergraph (aggregated graph).

Definition 2. An *A-hypergraph* (aggregated hypergraph) G is a graph the form:

$$G = (V, E, lab_{\{V,E\}}, att_{\{V,E\}}),$$

where V is a nonempty set of *super nodes*, such that $v \in V$ represents some S-hypergraph; $E \subset P_2(V)$ is a set of *external edges*, such that $\{u, v\} \in E$ represents the relationship between solids corresponding to S-hypergraphs given by super nodes u and v; $lab_\mu : \mu \to \mathcal{L}_\mu$, for $\mu = V, E$, is a labeling function for vertices and edges, respectively, with corresponding set of labels $\mathcal{L}_\mu$; $att_\mu : \mu \to 2^{\mathcal{A}_\mu}$ for $\mu = V, E$, is an attributing function for vertices and edges, respectively, with corresponding set of attributes $\mathcal{A}_\mu$.

A-hypergraphs and S-hypergraphs create a hierarchical structure: at the lower level, the detailed information about objects is stored (Figure 5); the upper level in turn carries information about mutual geometric relationships among objects: S-hypergraphs (Figure 6) are represented by *super nodes* and relations among them by graph edges and their attributes.

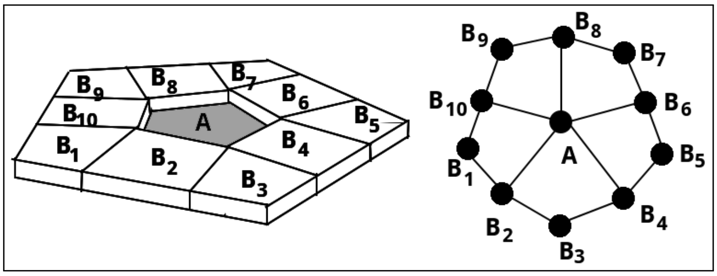


Figure 6. The Pentagon (left) as a loop of adhering solids (B_i's) with the internal area (A) and the corresponding aggregated hypergraph (A-hypergraph) (right).

In applications of the model, node and edge labels can be assigned with street names, luminaire identifiers, and so on. Attributes store operational data, such as the road width value, number of lanes, road surface reflectance, lighting class in the case of streets and pole GIS coordinates, fixture model and dimming value, pole height or spacing for luminaires (see Table 3).

Table 3. The sample attributes ascribed to graph edges and vertices.

Attribute Name	Description
bounding_box	Array of 2D GIS coordinates
lighting_class	Lighting class (as in EN 13201:2)
surface	Reflective properties of the road surface
luminaires	Array of relevant luminaires (together with all applicable data)

4.2. Slashed Graphs

As stated previously, the most appropriate representation for the considered systems is a graph-based one. Namely, we chose the *slashed graphs* approach introduced in [20]. In this graph distribution model, we decompose the main graph by *slashing* selected edges and adding pairs of virtual nodes as newly-created endpoints (see Figure 7).

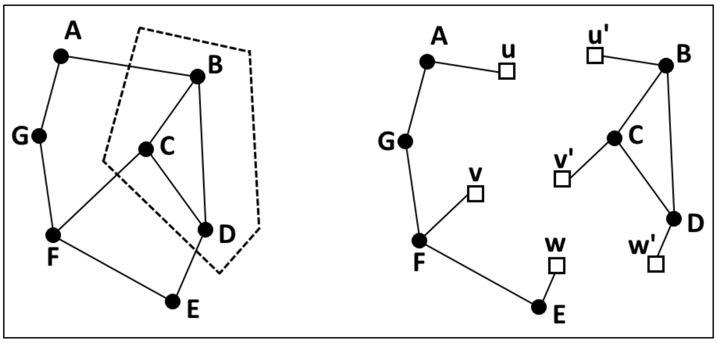


Figure 7. Centralized graph (left; the dashed line indicates the decomposition border) and its slashed form (right).

The formal definition of those notions is given below. For simplicity and without loss of generality, we abandon *att* functions in the sequel. Moreover, *lab* will stand for a pair of functions, namely lab_V and lab_E.

Definition 3 (Slashed form of G)**.** Let $\mathcal{G}$ be a family of directed A-hypergraphs, such that an edge structure is given by $E \subseteq V \times (\mathcal{L}_E \times 2^{\mathcal{A}_E}) \times V$, and let $G = (V, E, lab, att) \in \mathcal{G}$. A set $\{G_i\}$ of graphs is defined as follows.

- $G_i = (V_i, E_i, lab_i) \in \mathcal{G}$ and $V_i = C_i \cup D_i, C_i \cap D_i = \emptyset$, where C_i is a set of *core nodes*, D_i denotes a set of *dummy nodes* and $lab_i \equiv lab|_{V_i}$;
- $\bigcup_i C_i = V$ where C_i, C_j are mutually disjoint for $i \neq j$;
- $\forall v \in D_i \, \exists ! v' \in D_j (i \neq j)$, such that v' is the replica of v; $\forall v \in D_i \deg(v) = 1$;
- $\forall e \in E_i : \; e$ is incident to at most one dummy node.

An edge incident to a dummy node is called a *border edge*. The set of all border edges in G_i is denoted as E_i^b. A set $E_i^c = E_i - E_i^b$ is referred to as a set of *core edges* of G_i. Let $M \equiv \mathcal{L}_E \times 2^{\mathcal{A}_E}$, then a set $\{G_i\}$ as defined above is referred to as a *slashed form* of G and denoted $\not{\mathcal{G}}$, iff the following conditions are satisfied:

1. $\forall G_i^c = (C_i, E_i^c, lab_i|_{C_i})$, $\exists H_i \subset G : \; H_i \overset{\alpha}{\simeq} G_i^c$ (α denotes an isomorphic mapping between graphs), and H_i, H_j are disjoint for $i \neq j$.
2. $\exists f : M^2 \to M$, a bijective mapping satisfying: $\forall (e, e') \in E_i^b \times E_j^b \; (i \neq j)$, such that:

   (a) $e = (x_c, m, v) \in C_i \times M \times D_i$, $e' = (v', m', y_c) \in D_j \times M \times C_j$;
   (b) v' is a replica of v;

   $\exists ! \, e_{ij} = (x, m_e, y) \in E$ such that $x_c = \alpha(x), y_c = \alpha(y)$ and $f(m, m') = m_e$. e_{ij} is called a *slashed edge* associated with replicated dummy nodes v, v'.
3. $\forall e = (x, m, y) \in E$: (i) $\exists ! \, e_c \in E_i^c$ for some i, such that $e_c = \alpha(e)$; or (ii) $\exists ! \, (v, v') \in D_i \times D_j$ for some i, j, such that e is a slashed edge associated with v and v'.

   $G_i \in \not{\mathcal{G}}$ is called a *slashed component* of G.

Note that f mapping recovers the labeling/attributing data of a slashed edge on the basis of the labeling/attributing of relevant border edges.

In the context of the considered computational problem, a city map is represented by some A-hypergraph $G = (V, E, lab, att)$. Thus, the main task of decomposition is accomplished by transforming it to the form $\not{\mathcal{G}}$. The splitting algorithm is presented in [20]. Its computational complexity is $\mathcal{O}(|E|)$.

5. Agent System Perspective

The multi-agent system architecture for the considered computations relies on two agent types only: *master agent* (MA) and *computing agent* (CA).

The role of a master agent, which exists in a single instance only, is decomposing a problem-specific graph into a slashed form representing all sub-tasks to be processed. Moreover, an MA collects sub-results incoming from computing agents (Figure 8, left). Optionally, it can redeploy a CA to restart some computations when a failure (timeout) is detected. An MA has to ensure that all luminaires are covered by photometric computations.

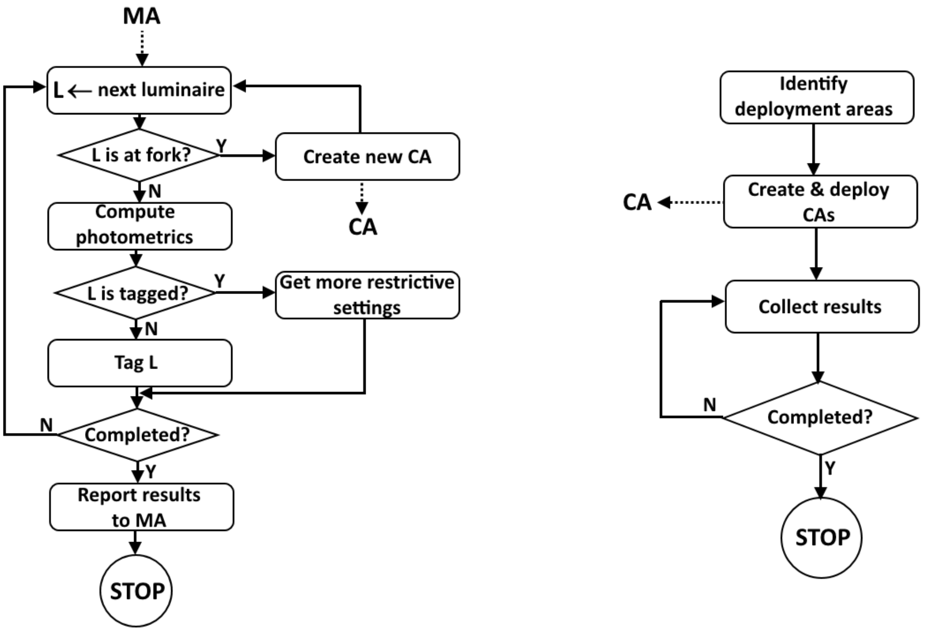


Figure 8. Multi-agent system activity schema. *Master agent* (MA) life cycle (left) and *computing agent* (CA) life cycle.

The detailed scenario of cooperation among a master agent and computing agents is the following. After deploying all CAs to their slashed components, an MA sends a CALC (calculate) request to all of them to start a design (optimization) process. As soon as it is completed, a CA submits a RESULT message to a master agent, containing the calculated output. It contains a full set of adjustments computed for all lighting installations falling into the relevant CA's area. Then, a CA is terminated. Note that an MA can track the overall optimization progress, so, optionally, if it lacks some results (due to any random failure), it can re-instantiate a given CA to complete a missing operation.

The computing agent that exists in multiple instances is responsible for executing a calculation processes as described in Section 2.2. In particular, a CA can create other CAs in "forking points". A CA life cycle terminates ("*Completed?*" testing block in Figure 8, right) in one of the following circumstances:

1. All non-tagged luminaires have been already processed. This happens when either a CA met a tagged luminaire or no other lamp to be processed is available in the relevant region.
2. No solution can be found. Such a situation occurs when a problem of finding an optimal adjustment complying with a given lighting class cannot be found in the assumed search space. In this case, a CA reports a problem to an MA. Then, a master agent may restart calculations in a changed search space (e.g., with other fixture models or pole heights).

Figure 9 shows the map with multiple segments being processed by computing agents.

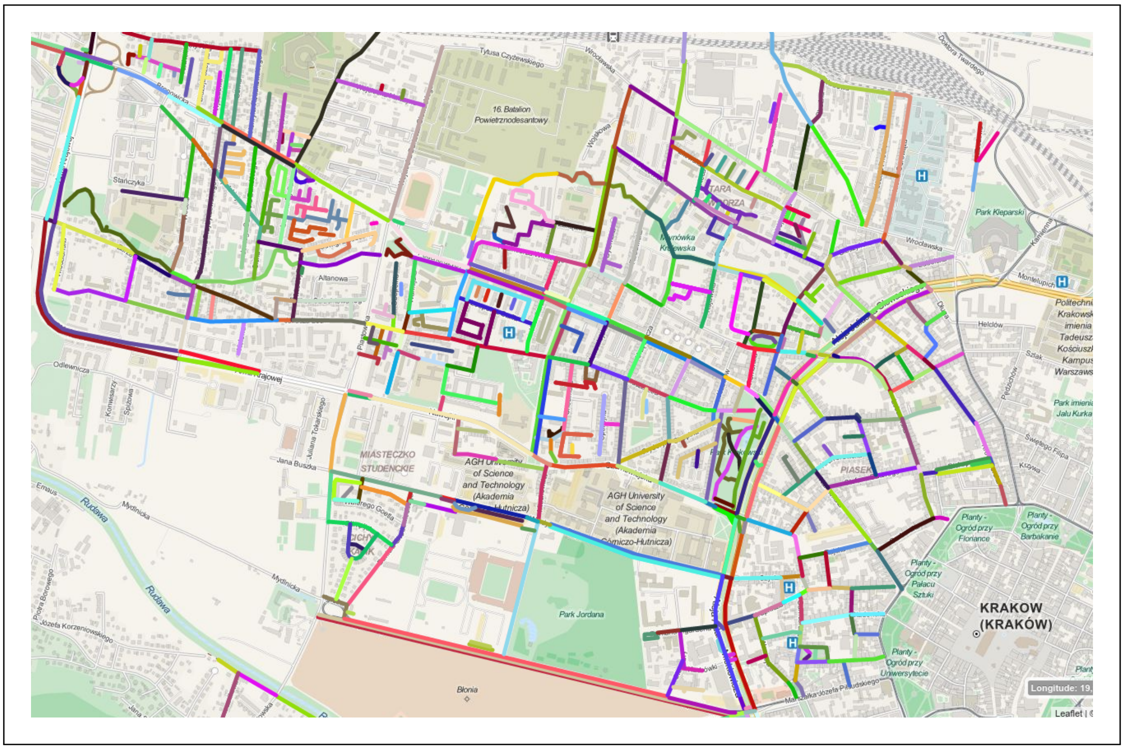

16. Batalion Powietrznodesantowy
Stańczyka
Altanowa
Odlewnicza
Jana Buszka
MIASTECZKO STUDENCKIE
AGH University of Science and Technology (Akademia Górniczo-Hutnicza)
Park Jordana
Błonia
Rudawa
Mydlnicka
Park Kleparski
Wrocławska
Politechnika Krakowska imienia Tadeusza Kościuszki Kampus Warszawska
PIASEK
Planty - Ogród przy Florianie
Planty - Ogród przy Barbakanie
Planty - Ogród przy Pałacu Sztuki
Planty - Ogród przy Uniwersytecie
KRAKOW (KRAKÓW)
Longitude: 19.
Leaflet | ©

Figure 9. The sample map with segments processed by agents.

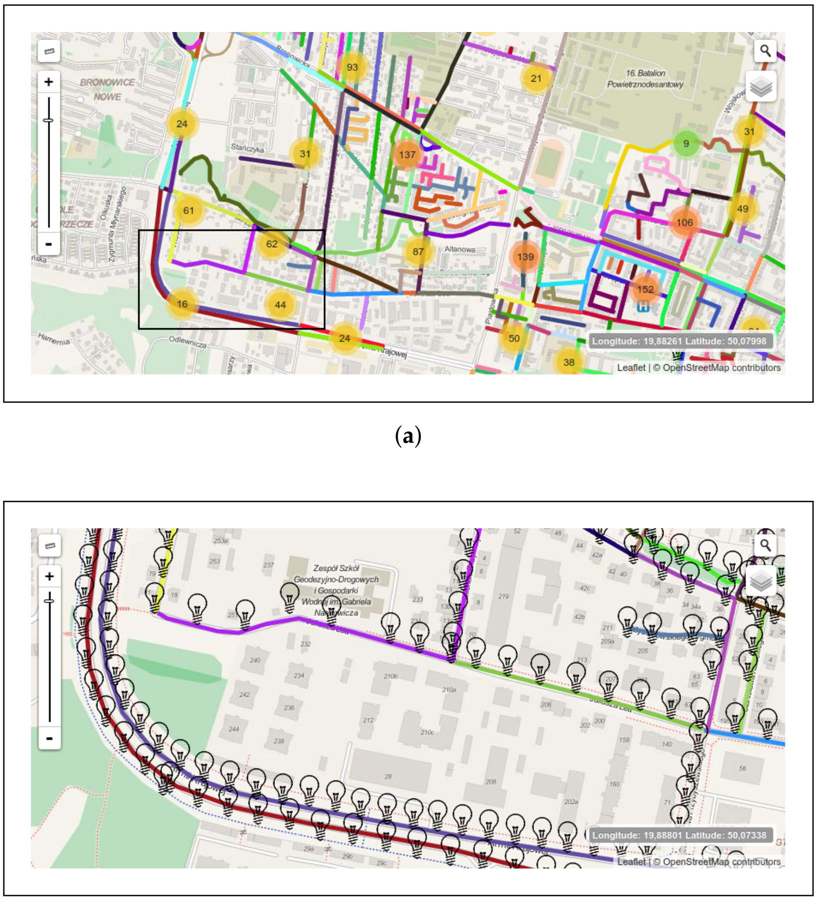
The relation between a multi-agent system and an underlying graph structure has to be mentioned. A master agent, which is responsible for decomposing the upper level of a hypergraph hierarchy, operates initially on an A-hypergraph being subject to a decomposition process. After completing A-hypergraph decomposition, creating CAs and assigning them to particular slashed components, an MA no longer possesses a graph. Instead, the CAs are activated on slashed components of a distributed A-hypergraph. Those components are expanded by CAs to S-hypergraphs. Note that a fully-expanded, hypergraph representation of a space enables describing such phenomena as a light being reflected from buildings' walls, falling onto a roadway surface and, thus, contributing to its illumination.

6. Results

The proposed approach was applied to support the retrofit of lighting installations located alongside 239 streets in the city of Kraków, Poland. The goal of the project was replacing 3700 HID (high-intensity discharge) fixtures with LED sources. Prior to that, the optimization of adjustments was performed for all of those luminaires. During that, 2000 fixture models (precisely, this number refers to the photometric solids) provided by several vendors were tested. Selected fixtures had to be matched appropriately to the area type (walkway, main road, park, parking zone, and so forth). Besides a fixture's model, the other parameters, such as the inclination angle and dimming, were also varied while searching for optimal configurations.

Due to the non-homogeneous geometry, the considered streets were logically subdivided into over 600 sections having uniform layouts. The most common road layout contained a single carriageway with walkways on both sides (see Figure 3). Note that each of them had a separate lighting class ascribed. In terms of typical photometric computations, the project covered approximately $3 \times 600 = 1800$ calculation fields. Each of those fields is represented in calculations as a hypergraph's node.

Figure 10a presents a high-level view on several streets being processed (the OpenStreetMap underlay is used). Its zoomed section is shown in Figure 10b: single luminaires are marked as bulbs.


BRONOWICE NOWE
Stańczyka
Altanowa
16. Batalion Powietrznodesantowy
93
21
24
31
137
9
31
61
49
106
62
87
139
16
44
152
24
50
38
Longitude: 19,88261 Latitude: 50,07998
Leaflet | © OpenStreetMap contributors
Zespół Szkół Geodezyjno-Drogowych i Gospodarki Wodnej im. Gabriela Narutowicza
Longitude: 19,88801 Latitude: 50,07338
Leaflet | © OpenStreetMap contributors

(**a**)

(**b**)

Figure 10. Computing task from the global and local perspective. (**a**) Global view on a computing task form an MA's perspective. The numbers in circles denote the numbers of luminaires in particular subdomains. The black rectangle delimits the area enlarged in (**b**); lighting installation layout from the perspective of a CA. Note that several CAs can operate in such a region.

Table 4 presents the results obtained for three methods of a lighting design. The first one is based on typical calculations as made by market-available software. The second one relies on standard computations, as well, but involves the optimization of adjustments, as well. In the third method, the customized approach is applied. Namely, the optimization is performed on the GIS-based street layout and luminaire location data according to the algorithm presented in [6].

Table 4. Resultant installation power obtained in different computation approaches.

Computing Method	Power Required (kW)	Power Savings *
No optimization, standard calculations	275	0.0%
Optimized in standard calculations	253	8%
Optimized in customized calculations	234	15%

* Compared to non-optimized, standard calculations.

7. Conclusions

The presented calculating method combines the preparation of well-suited photometric designs yielding significant energy savings and scalable agent-based computations, which allows for a time-efficient search over the entire solution space. It defines a multi-agent system framework, which was successfully applied in real-life retrofit projects covering thousands of luminaires.

Computing agents involved in the above operations are responsible for optimal solution finding and local optimizations in "contact zones", where two or more lamp series meet.

The introduced methodology allows not only for optimizing roadway lighting installations, but also enables the application of adaptive control methods, which fit roadway lighting performance to the actual requirements. These control methods rely on dynamic schemas where lamp dimming is set on the basis of an actual environment state reported by the sensor layer (induction loops, ambient light meters, rain sensors, and so forth). The role of the presented method is pre-optimization of all possible states of an installation. It reduces the power usage up to 15% when relying on the automated optimization and customized computation approach.

Acknowledgments: The paper publication costs were covered from the resources of the FOGA (Long life interconnected smart battery system for off-grid applications) project, grant no. 7.7.120.7050.

Author Contributions: Adam Sędziwy prepared graph and entropy related sections. Leszek Kotuski described the agent system architecture and performance (design process). Both authors have read and approved the final manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Outdoor Street Lighting. Available online: http://goo.gl/mfpGYT (accessed on 25 February 2016).
2. Rabaza, O.; Peña-García, A.; Pérez-Ocón, F.; Gómez-Lorente, D. A simple method for designing efficient public lighting, based on new parameter relationships. *Expert Syst. Appl.* **2013**, *40*, 7305–7315.
3. Gómez-Lorente, D.; Rabaza, O.; Estrella, A.E.; Peña-García, A. A new methodology for calculating roadway lighting design based on a multi-objective evolutionary algorithm. *Expert Syst. Appl.* **2013**, *40*, 2156–2164.
4. Wojnicki, I.; Ernst, S.; Kotulski, L.; Sędziwy, A. Advanced street lighting control. *Expert Syst. Appl.* **2014**, *41*, 999–1005.
5. Salata, F.; Golasi, I.; Bovenzi, S.; Vollaro, E.D.L.; Pagliaro, F.; Cellucci, L.; Coppi, M.; Gugliermetti, F.; Vollaro, A.D.L. Energy Optimization of Road Tunnel Lighting Systems. *Sustainability* **2015**, *7*, 9664–9680.
6. Sędziwy, A. A New Approach to Street Lighting Design. *LEUKOS* **2015**, doi:10.1080/15502724.2015.1080122.
7. Sędziwy, A.; Kozień-Woźniak, M. Computational Support for Optimizing Street Lighting Design. In *Complex Systems and Dependability*; Springer: Berlin/Heidelberg, Germany, 2012; Volume 170, pp. 241–255.
8. Temkin, O.N.; Zeigarnik, A.V.; Bonchev, D.B. *Chemical Reaction Networks: A Graph-Theoretical Approach*; CRC Press: Boca Raton, FL, USA, 1996.
9. Ramadan, E.; Tarafdar, A.; Pothen, A. A hypergraph model for the yeast protein complex network. In Proceedings of the 18th International Parallel and Distributed Processing Symposium, Santa Fe, NM, USA, 26–30 April 2004.
10. Pimm, S.L. Food webs. In *Food Webs*; Springer: Berlin/Heidelberg, Germany, 1982; pp. 1–11.
11. Kotulski, L.; Sedziwy, A.; Strug, B. Heterogeneous graph grammars synchronization in CAD systems supported by hypergraph representations of buildings. *Expert Syst. Appl.* **2014**, *41*, 990–998.
12. Kotulski, L.; Sedziwy, A. Parallel Graph Transformations Supported by Replicated Complementary Graphs. Adaptive and Natural Computing Algorithms; Dobnikar, A., Lotrič, U., Šter, B., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; Volume 6594, pp. 254–264.
13. DIALux. Available online: http://goo.gl/21XOBx (accessed on 25 February 2016).
14. Road lighting—Part 3: Calculation of performance. Available online: https://www.fer.unizg.hr/_download/repository/en_13201-3.pdf (accessed on 25 February 2016).
15. Road Lighting—Part 2: Performance Requirements. Available online: https://www.fer.unizg.hr/_download/repository/en_13201-2_.pdf (accessed on 25 February 2016).
16. Sedziwy, A.; Kotulski, L. A new approach to power consumption reduction of street lighting. In Proceedings of the International Conference on Smart Cities and Green ICT Systems (SMARTGREENS), Lisbon, Portugal, 22–25 May 2015; pp. 1–5.

17. *Handbook of Graph Grammars and Computing by Graph Transformations, Volume 1: Foundations*; Rozenberg, G., Eds.; World Scientific: Singapore, Singapore, 1997.
18. *Handbook of Graph Grammars and Computing by Graph Transformation, Volume 2: Applications, Languages and Tools*; Ehrig, H., Engels, G., Kreowski, H.J., Rozenberg, G., Eds.; World Scientific: Singapore, Singapore, 1999.
19. Kotulski, L.; Strug, B. Using Graph Transformations in Distributed Adaptive Design System. In *Computer Vision and Graphics*; Bolc, L., Kulikowski, J.L., Wojciechowski, K., Eds.; Springer: Berlin/Heidelberg, Germany, 2008; Volume 5337, pp. 477–486.
20. Sędziwy, A. Effective Graph Representation for Agent-Based Distributed Computing. In *Agent and Multi-Agent Systems. Technologies and Applications*; Jezic, G., Ed.; Springer: Berlin Heidelberg, Germany, 2012; Volume 7327, pp. 638–647.
21. Kotulski, L.; Sędziwy, A. GRADIS—The multiagent environment supported by graph transformations. *Simul. Model. Pract. Theory* **2010**, *18*, 1515–1525.
22. Sędziwy, A. Effective graph representation supporting multi-agent distributed computing. *Int. J. Innov. Comput. Inf. Control* **2014**, *10*, 101–113.