Graph-Based Optimization of Energy Efficiency of Street Lighting

Adam Sędziwy(✉) and Leszek Kotulski

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

Abstract. Designing energy efficient street lighting got an emerging domain due to technological potential of LED light sources. To profit those solutions however one has to use effective computational methods enabling low-energy-solution finding. In this article we focus particularly on the problem of discovering and removing over-illuminated areas being the side-effect of using typical lighting design methods, especially in the case of non-regular areas. We propose the approach which combines hypergraph-based modeling of objects, the concept of *slashed graphs* and heuristics addressing optimal lighting issue. The synergy of those three parts creates practically usable methodology for time and energy efficient outdoor lighting design.

Keywords: Slashed graphs · Street lighting · Agent systems · Bulk computations

1 Introduction

The emergence of advanced, LED-based technologies on the outdoor lighting market opened new perspectives on creating energy efficient and well suited lighting infrastructures [18]. Two most important drivers staying behind are: luminous flux dimming capability (which was rudimentarily present for high-intensity discharge lamps, abbrev. HID) and possibility of *shaping* a photometric solid of a fixture [6,9]. In particular, the former property is not reachable in other technologies: dimming HID or metal halide lamps is not feasible in the full range (i.e., 0-100%) and reduces the source's lifetime. Besides that, the onset time of typical light sources is significantly higher than for LEDs, and makes those sources unsuitable for an adaptive street lighting.

A lot of commercial photometric software profiting of those properties are available on the market. Their common feature however is ability of making computations for a single scene only. Thus, if one aims at computing photometry for some region consisting of several blocks and including a couple of streets then he needs to set up manually a sequence of separate scenes corresponding to particular lighting situations. Another problem arising in this context is computing photometry and optimizing street lighting configuration for non-typical (contrary to straight, uniform street sections) road situations like cross roads,

L. Rutkowski et al. (Eds.): ICAISC 2015, Part II, LNAI 9120, pp. 515–526, 2015.
DOI: 10.1007/978-3-319-19369-4_46

conflict areas and so on. The typical approach to photometric design results in producing over-illuminated scenes and thus increased energy usage (see [2]).

Summarizing the issues mentioned above: there is no tool enabling time efficient and scalable photometric computing, especially for non-standard scenes. To illustrate the problem of a time efficient solution finding in lighting design tasks let us consider the trivial example of a row of ten LED lamps[1], being designed along some motor road. Since the considered installation is made from scratch the following parameters may be varied to find the best solution:

1. fixture model/photometric solid (500 – variants),
2. fixture mounting angle (from 0° to 20° with step 1° – 21 variants),
3. pole height (from $6m$ to $12m$ with step $0.1m$ – 61 variants),
4. lamp spacing (from $20m$ to $40m$ with step $0.1m$ – 201 variants),
5. arm length (from $0m$ to $2m$ with step $0.1m$ – 21 variants),
6. dimming level (from 0% to 99% with step 1% – 100 variants).

The resultant total number of variants to be checked is $N \approx 2.7 \times 10^{11}$.

The computing time for a single variant takes about 0.1 s, so the total time of the calculation (for all variants) takes over 14 years on a single 4-core processor. What is worse, in real life cases we design photometric scenes containing thousands of lamps (e.g., about 7,000 lamps in SOWA Project carried out in Cracow, Poland). We suggest to solve the above problem twofold: by applying AI methods to finding an optimal variants and by parallelization of computations.

In this paper the representation of slashed graphs [21] as a formal background for this purpose is proposed. It offers a flexible and scalable formal framework for scene modeling, allowing parallelizing and distributing the photometric computations. It enables defining a set of computational tasks ascribed to subsequent lighting situations covering the entire considered area. The proposed methodology is based on a problem's graph representation which is a broadly used in various domains [15,17,14,7,8,11,19]. Such a graph model constitutes an environment for a multi-agent system. Particular agents are ascribed to subtasks performed in parallel and an agent's knowledge is sewn in a graph structure [16,12,13].

The key property of this methodology is processing a given area not "as is" i.e., with some assigned set of luminaires but analyzing all lamps available in its neighborhood, selecting those among them which influence a scene and, in the sequel, dividing logically a scene into subareas having different levels of illumination. The objective is to equalize those levels, reducing power usage overhead and stay aligned with mandatory lighting standards. Such an approach allows for more accurate adjustment of lamp work parameters (dimming, photometric solid) and thus minimizing the power consumption.

Although a single task (i.e., made on one scene) seems to be a low complexity operation, from the perspective of an entire considered urban space however we face the scalability related issue. To overcome this problem we use *slashed*

[1] Computing time may be regarded as constant wrt the number of poles.

graphs model introduced in [21] This formalism was introduce to support parallel distributed computations.

The structure of the article is organized as follows. In the next section the problem formulation will be introduced. Sections 3 and 4 contain formal basics of used graph models: the short overview of the used hypergraph representation of complex solids and the slashed graphs model mentioned above. The way of formulating a computational task is presented in Section 5 while an overall computation process will be discussed in Section 6. The final conclusion may be found in Section 7.

2 Problem Formulation

The main objective of the outdoor lighting design is preparing infrastructure which fulfills mandatory standards ([3,4,10]) related to lamps performance. Those norms are expressed quantitatively by threshold values of certain photometric quantities which have to be ensured by a lighting system. Among those quantities are: average illuminance – E_{avg}, average luminance – L_{avg}, longitudinal and overall uniformity – U_l and U_o respectively, threshold increment – TI, surround ratio – SR and others. Values of those parameters for each a road must not cross the established thresholds defined by a standard. The sample set of norms given for street lighting is shown in Table 1.

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

class	L_{avg} $[cd/m^2]$ [min. value]	U_o [min. value]	U_l [min. value]	TI [%] [max. value]	SR [min. value]
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	-

When designing lighting for a zone consisting of numerous streets and pedestrian routes one has to perform computations for multiple lighting situation (LS) assigned to particular areas. Each LS is defined for a single area A which may be either homogeneous in terms of geometric properties (number of lanes, width, surface type) and street lamps locations (arrangement, spacing, setback, overhang) or non-regular e.g., for some types of road junctions (see Fig.1). Applying the results reusing approach in the former case [22] may reduce significantly the computing time. Another factor which has to be taken into account while considering an LS, is the presence of objects (buildings) dropping shadows on it.

In those circumstances some luminaires may be excluded from computations as not affecting a scene.

Figure 1 shows the trivial example of a scene: the cross road (area A_1) with three incoming streets (areas A_2, A_3, A_4) and adjacent sidewalks. Luminaires are marked with diamonds. Moreover, circles indicating light ranges are shown for selected lamps[2]. A particular lighting situation, delimited with a dashed line, consists of a relevant area and luminaires located on it. It should be stressed that a given area may be affected by luminaires belonging to foreign LS's, for example lamps 2 and 3 standing on A_2 influence illuminance on A_1.

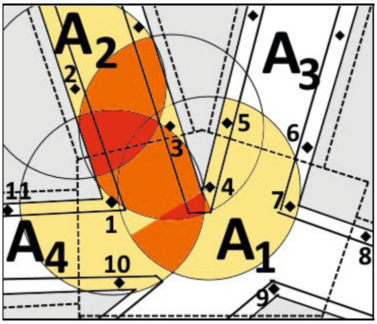

Fig. 1. The sample crossroad. The diamond marks denote luminaires and gray areas - buildings. For the better readability the figure illustrates an impact of four lamps only (numbered $1 \dots 4$) and the lamps are denoted with indices only (instead of F_i). A circle delimits a range of a single luminaire while colors are related to overlapping of particular ranges: *red* – 3 lamps, *orange* – 2 lamps, *yellow* – single luminaire

The complete input for photometric computations performed for a given LS, is a compound of road related data ($\mathbf{R}_d$) and luminaires related data ($\mathbf{L}_d$). The latter ones include a set of fixtures which luminous flux affects a given area. This set will be denoted as $\text{Aff}(A)$ where A is an area underlying LS. For Figure 1 we have: $\text{Aff}(A_1) = \{F_1, F_2, \dots F_{11}\}$. The impact of other fixtures is neglectable due to actual distances.

3 Graph Model of Urban Space

To enable the efficient analysis and modeling of large-scale or complex systems, their description needs to be formalized. Such a formalization is required prior to computer aided problem solving. In the paper [20] the 3D-compliant hypergraph model of an urban space was introduced. Hypergraph formalism is broadly accepted and used in the 3D modeling [5,1,24]. In this model both buildings and areas (e.g. streets, sidewalks, squares, lawns) are represented by hypergraphs[3]

[2] Note that a circle is valid only for a photometric solid having radial symmetry. We make here such a simplifying assumption for more clarity.

[3] In the case of surfaces a given area is assigned with an infinitesimal height to obtain 3D object which, in the sequel, may be described by a hypergraph.

with fixtures (luminaires) as their vertices (or more precisely, vertices of a certain type).

Besides hypergraphs, we will also use the typical graph model (see Section 4) as it will be obtained as a result of hypergraphs *aggregation*. The basics of the hypergraph approach and its transition to the regular graphs presented below.

Hypergraphs. Let T will be a polyhedron consisting of n faces $(f_1, f_2, \ldots f_n)$, k edges $(a_1, a_2, \ldots a_k)$ and m vertices $(h_1, h_2, \ldots h_m)$. Hypergraph H_T representing the polyhedron T is a tuple $H_T = (V, A, H)$, where $V = \{f_1, f_2, \ldots f_n\}$ is a set of hypergraph nodes (representing physical faces), $H = \{h_1, h_2, \ldots h_m\}$ is a set of hypergraph hyperedges (representing physical vertices), and $HA = \{a_1, a_2, \ldots a_k\}$ is a set of hypergraph edges (representing physical edges).

Figure 2 presents the hypergraph model of a cuboid. For the case of two adhering polyhedrons, we introduce the additional, logical type of edge (referred to as an *external edge*), which denotes adherence of two faces.

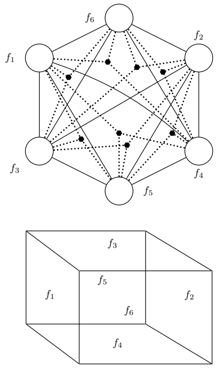

Fig. 2. A hypergraph representation of a cuboid

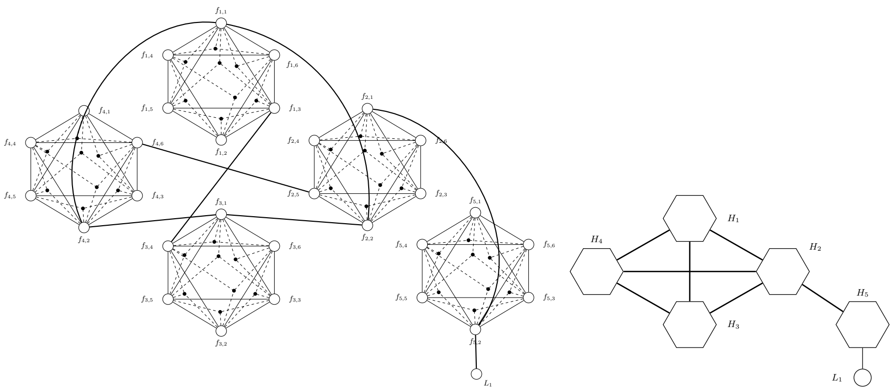

Fig. 3. A hypergraph model of adhering polyhedrons (left) and its aggregated form (right)

In some cases, one requires general information about an overall object structure rather than its detailed properties. In such circumstances, it is sufficient to get information about relations among particular components of a considered scene. To accomplish that, we transform (*collapse*) hypergraphs to vertices using *aggregating transformation*. Such an operation, also referred to as *synthesis*, is described in [23]. All data describing how external edges (i.e. edges connecting two adjacent solids) were attached to the collapsed hypergraph are stored as values of supplementary attributes of those edges. Thanks to that the inverse operation, namely *analysis*, may be executed [23].

Figure 3 (left) presents a hypergraph model of five adhering cuboidal objects (external edges are represented as thick lines). Additionally, one of them has a point object attached (which may represents fixture/luminaire). In Figure 3, the aggregated form of this composite object is presented: hexagonal nodes $(H_1, \ldots, H_5)$ correspond to *collapsed* hypergraphs.

4 Slashed Graphs

In this section a brief overview of a distributed graph representation will be presented. This model is crucial for parallel execution of photometric computations.

Definition 1. *$(\mathcal{L}, \mathcal{A})$-graph is a digraph $G = (V, E)$, where V is nonempty, finite set of indexed nodes, $E \subseteq V \times (\mathcal{L} \times \mathcal{A}) \times V$ is a set of arcs and $\mathcal{L}$, $\mathcal{A}$ denote respectively a set of labels and attributes of edges. The family of $(\mathcal{L}, \mathcal{A})$-graphs will be denoted as $\mathcal{G}$.*

In the sequel we introduce the notion of *slashed form* of a graph, being fundamental for $(\mathcal{L}, \mathcal{A})$-graph decomposition considered further.

Definition 2. *Let $G = (V, E) \in \mathcal{G}$. We define a set $\{G_i\}_{i \in I}$ of graphs in the following manner: (i) $G_i = (V_i, E_i) \in \mathcal{G}$ and $V_i = C_i \cup D_i$, $C_i \cap D_i = \emptyset$, where C_i is a set of **internal vertices** and D_i denotes a set of **dummy nodes**, (ii) $V = \bigcup_i C_i$, where $C_i \cap C_j = \emptyset$ for $i \neq j$, (iii) $\forall v \in D_i \, \exists ! v' \in D_j (i \neq j)$ such that v' is a replica of a node v; $\forall v \in D_i : d_{G_i^\star}(v) = 1$[4], (iv) $\forall e \in E_i : e$ is incident to at last one dummy node.*

*An edge incident to a dummy node will be referred to as a **border edge**. A set of all border edges in a graph G_i will be denoted as E_i^b. On the other side $E_i^c = E_i - E_i^b$ is a set of **internal edges** of G_i.*

*Let us denote $M = \mathcal{L} \times \mathcal{A}$ then $\{G_i\}$ defined above is a **slashed form** of G (denoted as $\not{G}$) iff following conditions are satisfied:*

1. *$\forall G_i^c = (C_i, E_i^c)$, $\exists H_i \subset G : H_i \overset{\alpha}{\simeq} G_i^c$ and H_i, H_j are disjoint for $i \neq j$ ($\overset{\alpha}{\simeq}$ denotes isomorphism α between graphs).*
2. *There exists bijection $f : M^2 \to M$ such that for $(e, e') \in E_i^b \times E_j^b \, (i \neq j)$, where $e = (x_c, m, v) \in C_i \times M \times D_i$, $e' = (v', m', y_c) \in D_j \times M \times C_j$ and v' is a replica of a node v, there exists exactly one edge $e_{ij} = (x, m_e, y) \in E$ such that $x_c = \alpha(x), y_c = \alpha(y)$ and $f(m, m') = m_e$. e_{ij} is referred to as a **slashed edge** associated with 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 dummy nodes v, v'.*

*$G_i \in \not{G}$ is referred to as a **slashed component** of a graph G.*

[4] For a given digraph G, $G^\star$ denotes an underlying undirected graph.

Having the formal definition of a graph's slashed form we may introduce two complementary mappings. $\Delta : \mathcal{G} \to \mathcal{G} \times \mathcal{G}$ decomposes $X \in \mathcal{G}$ into slashed components $X_1, X_2 \in \mathcal{G}$ and $\Psi : \mathcal{G} \times \mathcal{G} \to \mathcal{G}$ merges two slashed components and removes all relevant dummy nodes shared by both of them.

In Figure 4 the centralized and slashed form of a certain graph G are shown. Internal nodes are denotes as circles ($\bigcirc$), and dummy ones as squares ($\square$). Moreover the following indexing convention is applied (see Figure 4b).

- An index of an internal node has the form (i, k), where i is an unique identifier of a slashed component $G_i \in \not{\mathcal{G}}$ and k is an unique (in G_i) node index.
- A dummy node index has the form $(-1, k)_r$, where k is a globally unambiguous identifier of a pair of dummy nodes and r is a reference of a slashed component containing the complementary replica of a given dummy node. Using reference r enables instant retrieving of a dummy node's replica.

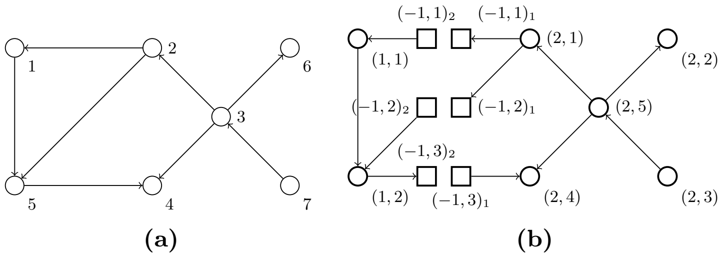

Fig. 4. (a) Graph G (b) $\not{\mathcal{G}}$ representation

The above definition of a slashed representation may be too complicated for practical use. For that reason two transformations, namely Split and Merge, enabling migration between centralized and distributed graph model were introduced in [21]. The complexity of both operations is $\mathcal{O}(|E|)$ which means that it is linear wrt a number of dummy node pairs. The detailed discussion on their complexity in a distributed environment may be found in the mentioned article.

5 Defining Computational Tasks

Before the algorithm of LS's generation will be introduced let us define the following auxiliary notation.

1. V_{com} denotes a set of nodes corresponding to computational areas like streets with sidewalks, cross sections, conflict areas and so on.
2. $N(x \in V_{agg})$ is a set of aggregated nodes (i.e., representing some hypergraphs) being the neighbors of x.
3. L_x for $x \in V_{agg}$ is a set of nodes labeled by F (i.e., representing fixtures) which are neighbors of x.
4. $B(u, v) = \{x : lab(x) = B \wedge x \in N(u) \cap N(v)\}$ for $(u, v) \in V_{agg} \times V_{agg}$ denotes a set of vertices representing buildings (labeled with B) which are adjacent to areas represented by u and v.

5. $B(u) = \bigcup_{v \in N(u)} B(u, v)$ is a set of all buildings adjacent to area u and to some area neighboring u.

Suppose that an area A represented by $v_A \in V_{com}$ is given. To perform photometric computations on A one has to collect all luminaires affecting it i.e., $\mathrm{Aff}(A)$. To accomplish that a neighborhood $N(v_A)$ has to be determined by using some search algorithm, e.g., BFS. Next, the influence of buildings adjacent to A on its illumination has to be investigated: for each $v \in N(v_A)$ we compute a set $L_v^- = \{x : lab(x) = F \wedge B(v_A)$ blocks the influence of F on $A\}$. Finally we get $\mathrm{Aff}(A) = \bigcup_{v \in N(v_A)} (L_v - L_v^-)$.

An important phase of computations leading to definition of an LS is related to evaluating the expression $B(v_A)$ *blocks the influence of* F *on* A, present in the definition of L_v^-. This evaluation is accomplished by geometric analysis of an overlapping of physical objects and photometric solids.

Next step is subdividing an LS's underlying area, A, into segments on the basis of illuminance levels on A. The example is demonstrated in Figure 5: the lighting situation is bounded by the thin dashed line and segments obtained by an area subdivision are separated by the thick dotted one. Each single luminaire is associated with a circular range limit (filled with light yellow). Zones where ranges overlap are colored with either yellow (overlapping of 2 ranges) or orange (3 ranges). There may be distinguished three subareas of A, having different average lighting levels (LL). Those subareas are denoted as A_1 (moderate LL), A_2 (low LL), A_3 (high LL).

Remark. Let us note that boundaries separating subareas are set in an arbitrary manner. In particular, to avoid too high granularity some lower limit for a subarea's size should be established.

The important property of the proposed approach is that lighting levels are tuned separately (for A_1, A_2, A_3) rather than for the entire area A. Thanks to this over-illuminating and related energy costs may be avoided. Prior to that $\mathrm{Aff}(A_i)$ sets have to be determined:

- $\mathrm{Aff}(A_1) = \{F_4, F_5, F_6, F_{11}\}$, $A_7 \notin \mathrm{Aff}(A_1)$ because A_1 is shaded by B_1,
- $\mathrm{Aff}(A_2) = \{F_1, F_4, F_6, F_7\}$,
- $\mathrm{Aff}(A_3) = \{F_1, F_2, F_3, F_4, F_8, F_9, F_{10}\}$.

Note that complexity of determining $\mathrm{Aff}(A_i)$ is polynomial with respect to both a number of luminaires and a number of streets incoming to A.

Finally we obtain:

$$LS_i = \{\mathbf{R}_d^{(i)}, \mathbf{L}_d^{(i)}\}, \text{ where } \mathbf{R}_d^{(i)} = A_i, \mathbf{L}_d^{(i)} = \mathrm{Aff}(A_i).$$

Tuning of a luminaire relies on adjusting its luminous flux. This operation does not change a radius of a range circle itself (or in the real-life cases – a shape and size of a range area) because photometric solid may be alternated by modifying a fixture's optics only [6,9]. One can decrease however intensity (by dimming) and thereby practically minimize an illuminated zone. Assuming that A_1 illuminance is compliant with a standard, fluxes from F_1, F_4 and F_7 should be increased while fixtures F_2, F_3, F_{10} have to be dimmed.

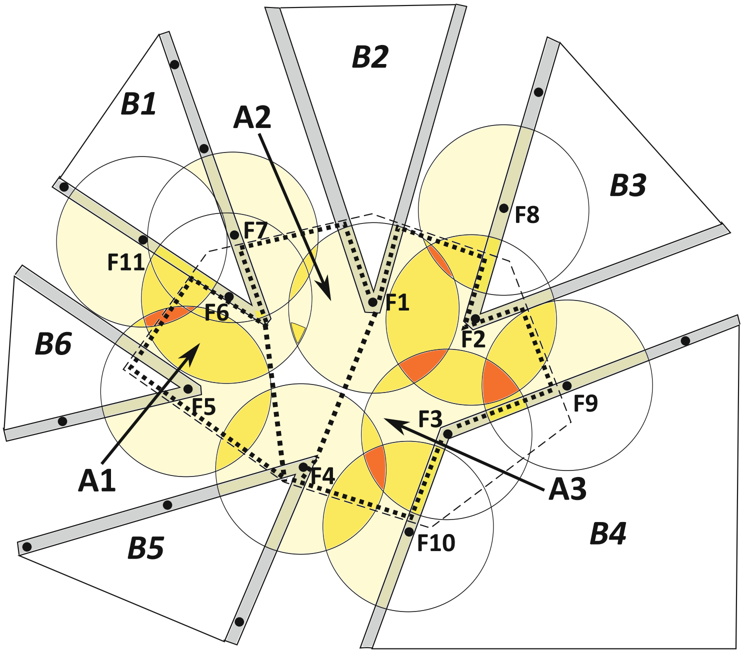

Fig. 5. The sample crossroad. Bolded points denote lamps and gray strips - sidewalks. Buildings are denoted with $B_1 \ldots B_6$ and considered luminaires with $F_1 \ldots F_{10}$. A circle delimits a range of a single luminaire while colors are related to overlapping of particular ranges: *orange* – 3 lamps overlapping, *yellow* – 2 lamps overlapping, *light yellow* – single luminaire range

6 Computations

Computational tasks introduced in the previous section are preformed on all aggregated nodes (see Fig.2) representing roads, sidewalks and other similar areas. Additionally, nodes representing buildings and other massive objects may be applicable during computations (see points 4 and 5 in Section 5). From the global perspective computations are made on a *normal* graph (i.e., not on a hypergraph) and they are node-oriented which means that almost all input data are located in aggregated vertices. For that reason a natural environment for their execution are slashed graphs introduced in Section 4.

From the local (aggregated node) perspective which covers blocks 3–5 in Figure 6, task execution may be either centralized (for simple, homogeneous lighting situations) or recursively distributed when a given area is partitioned (Section 2). For the latter case a slashed graph representation is used again: subsequent subareas are assigned with aggregated nodes hosting particular sub-calculations.

Regarding the polynomial complexities of subsequent phases of computations, the overall complexity is polynomial as well.

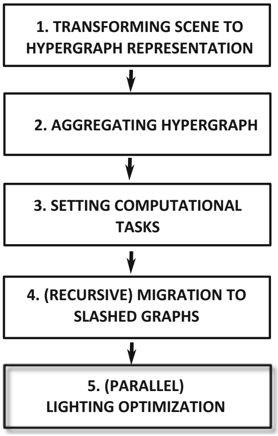

Fig. 6. Subsequent phases of computations

7 Summary

Well designed lighting infrastructure based on LED lamps enables achieving important energy savings in outdoor lighting. The rapid technological development in this field is not supported however by suitable tools capable of performing bulk computations, required in large scale retrofit projects (e.g., in cities) covering a huge number of streets, squares and so on. In previous articles (e.g., see [22]) the main effort was put into facilitating massive outdoor lighting optimization made on typical regular road situations. In this paper we fill this gap by introducing the enhancement to graph-based photometric computations, which enables optimizing lighting parameters for non-standard areas (e.g. non-regular crossroads). This goal is achieved by fine-grained analysis of a considered area and finding a balance between energy efficiency and requirements imposed by mandatory lighting standards.

The complexity of computations performed in the proposed methodology is polynomial but their execution in a distributed environment of slashed graphs reduces the computation time substantially. Computations performed in the SOWA project (7,000 lamps with altered parameters: pole height, arm length, fixture model, dimming) were completed in 6 hours.

References

1. Ansaldi, S., De Floriani, L., Falcidieno, B.: Geometric modeling of solid objects by using a face adjacency graph representation. SIGGRAPH Comput. Graph. 19(3), 131–139 (1985), http://doi.acm.org/10.1145/325165.325218
2. Boyce, P., Hunter, C., Vasconez, S.: An evaluation of three types of gas station canopy lighting. Tech. rep., Lighting Research Center, Rensselaer Polytechnic Institute, Troy, NY 12180-3352 (December 2001)
3. British Standards Institution (BSI): Road lighting. performance requirements, bs en 13201-2:2003, London: BSI (2003)

4. Commission Internationale de l'Eclairage: Lighting of Roads for Motor and Pedestrian Traffic, CIE 115:2010, Vienna: CIE (2010)
5. De Floriani, L., Falcidieno, B.: A hierarchical boundary model for solid object representation. ACM Trans. Graph. 7(1), 42–60 (1988), http://doi.acm.org/10.1145/42188.46164
6. Doskolovich, L.L., Dmitriev, A.Y., Bezus, E.A., Moiseev, M.A.: Analytical design of freeform optical elements generating an arbitrary-shape curve. Appl. Opt. 52(12), 2521–2526 (2013), http://ao.osa.org/abstract.cfm?URI=ao-52-12-2521
7. Ehrig, H., Heckel, R., Korff, M., Löwe, M., Ribeiro, L., Wagner, A., Corradini, A.: Algebraic approaches to graph transformation II: Single pushout approach and comparison with double pushout approach. In: Rozenberg, G. (ed.) The Handbook of Graph Grammars and Computing by Graph Transformation. Foundations, vol. 1. World Scientific (1997)
8. Engelfriet, J., Rozenberg, G.: Node replacement graph grammars. In: Rozenberg, G. (ed.) Handbook of Graph Grammars and Computing by Graph Transformation, vol. I, ch. 1, pp. 1–94. World Scientific (1997)
9. Feng, Z., Luo, Y., Han, Y.: Design of led freeform optical system for road lighting with high luminance/illuminance ratio. Opt. Express 18(21), 22020–22031 (2010), http://www.opticsexpress.org/abstract.cfm?URI=oe-18-21-22020
10. Illuminating Engineering Society of North America (IESNA): American National Standard Practice For Roadway Lighting, RP-8-00, New York: IESNA (2000)
11. Dasso, J.: w., Păun, G., Rozenberg, G.: Grammar systems. In: Salomaa, A., Rozenberg, G. (eds.) Handbook of Formal Languages, vol. 2, ch. 4, pp. 155–213. Springer, Heidelberg (1997)
12. Kotulski, L.: Distributed graphs transformed by multiagent system. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS (LNAI), vol. 5097, pp. 1234–1242. Springer, Heidelberg (2008)
13. Kotulski, L.: GRADIS – Multiagent Environment Supporting Distributed Graph Transformations. In: Bubak, M., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2008, Part III. LNCS, vol. 5103, pp. 644–653. Springer, Heidelberg (2008)
14. Kotulski, L., Strug, B.: Multi-agent system for distributed adaptive design. Key Engineering Materials 486, 217–220 (2011)
15. Kotulski, L., Sędziwy, A., Strug, B.: Heterogeneous graph grammars synchronization in {CAD} systems supported by hypergraph representations of buildings. Expert Systems with Applications 41(4, pt. 1), 990 – 998 (2014), http://www.sciencedirect.com/science/article/pii/S0957417413005307
16. Kotulski, L., Sędziwy, A.: GRADIS - The multiagent environment supported by graph transformations. Simulation Modelling Practice and Theory 18(10), 1515–1525 (2010)
17. Kotulski, L., Strug, B.: Distributed adaptive design with hierarchical autonomous graph transformation systems. In: Shi, Y., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2007, Part II. LNCS, vol. 4488, pp. 880–887. Springer, Heidelberg (2007)
18. Ramadhani, F., Bakar, K., Shafer, M.: Optimization of standalone street light system with consideration of lighting control. In: 2013 International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE), pp. 583–588 (May 2013)
19. Rozenberg, G., Ehrig, H., et al. (eds.): Handbook on Graph Grammars and Computing by Graph Transformation 3 (Concurrency). World Scientific, Singapore (1999)

20. Sędziwy, A.: Representation of objects in agent-based lighting design problem. In: Zamojski, W., Mazurkiewicz, J., Sugier, J., Walkowiak, T., Kacprzyk, J. (eds.) Complex Systems and Dependability. AISC, vol. 170, pp. 209–223. Springer, Heidelberg (2012)
21. Sędziwy, A.: Effective graph representation for agent-based distributed computing. In: Jezic, G., Kusek, M., Nguyen, N.-T., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2012. LNCS, vol. 7327, pp. 638–647. Springer, Heidelberg (2012)
22. Sędziwy, A.: On acceleration of multi-agent system performance in large scale photometric computations. In: Barbucha, D., Le, M.T., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA. Frontiers in Artificial Intelligence and Applications, vol. 252, pp. 58–67. IOS Press (2013)
23. Sędziwy, A.: Graph-Based Computing Environment For Parallel Computations And Its Application To Lighting Design Problem. Wydawnictwo AGH (2015) (in press)
24. Strug, B., Grabska, E., Ślusarczyk, G.: Supporting the design process with hypergraph genetic operators. Advanced Engineering Informatics 28(1), 11–27 (2014), http://www.sciencedirect.com/science/article/pii/S1474034613000785