Advances in Engineering Software 108 (2017) 57–67

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

A novel resource sharing algorithm based on distributed construction for radiant enclosure problems

Peter Finzell, Kenneth M. Bryden*

Simulation Modeling and Decision Science Program, Ames Laboratory, Iowa State University, 1620 Howe Hall, Ames, IA, 50011, USA

ARTICLE INFO

Article history:
Received 28 September 2016
Revised 7 December 2016
Accepted 26 February 2017
Available online 6 March 2017

Keywords:
Stigmergy
Multi-agent systems
Bio-inspired
Inverse heat transfer
Distributed construction

ABSTRACT

This paper demonstrates a novel approach to solving inverse radiant enclosure problems based on distributed construction. Specifically, the problem of determining the temperature distribution needed on the heater surfaces to achieve a desired design surface temperature profile is recast as a distributed construction problem in which a shared resource, temperature, is distributed by computational agents moving blocks. The sharing of blocks between agents enables them to achieve their desired local state, which in turn achieves the desired global state. Each agent uses the current state of their local environment and a simple set of rules to determine when to exchange blocks, each block representing a discrete unit of temperature change. This algorithm is demonstrated using the established two-dimensional inverse radiation enclosure problem. The temperature profile on the heater surfaces is adjusted to achieve a desired temperature profile on the design surfaces. The resource sharing algorithm was able to determine the needed temperatures on the heater surfaces to obtain the desired temperature distribution on the design surfaces in the nine cases examined.

1. Introduction

Radiant enclosure problems are encountered in many design problems, for example, in annealing, industrial process ovens, and combustion chambers [11,22]. These are problems where radiant heat transfer from several *heater surfaces* is used to establish and maintain a specified temperature distribution over several *design surfaces*. A design surface could be any material in a manufacturing process that requires precise or uniform temperature control such as glass or metal. These types of problems are concerned with the geometric design of the enclosure or the correct heater surface inputs to produce a desired temperature profile along the design surface or both [27]. Radiant enclosure design problems require developing a detailed analysis model, which can then be used in the design process. In the case considered here, the design process involves selecting the needed inputs to the heater surfaces to achieve the desired temperature profile along the design surfaces.

Frequently in engineering design, the analysis problem to be solved is an inverse problem, in which the desired outcome is specified and the goal is to specify the inputs required to obtain the desired outcome, i.e. the design variables. For example, in many radiant enclosure design problems the temperatures of the design surfaces are specified and the temperatures of the heater surfaces need to be determined. Direct solutions to these inverse problems, (e.g., using the known design surface temperatures to determine the needed heater surface temperature) is not a tractable solution method because there may be multiple solutions that yield the desired design outcome, many or all of which may be physically infeasible, or there may be no solutions. These types of inverse problems are mathematically ill-posed [28]. A problem is considered well posed when the problem is unique, a solution exists, and the solution depends on the data [23,38]. The ill-posed nature of these problems makes them sensitive to errors, and small changes in the input may significantly change the output [33].

To overcome this, three approaches are generally taken: trial and error, optimization, and regularization. In trial and error the design engineer uses information from the system, known or desired constraints, and experience to make an educated guess about what inputs are needed to achieve the desired design. That educated guess can be then iteratively refined to achieve the desired design. However, in many cases trial and error is too time consuming and optimization or regularization are used to find the needed design inputs [25]. Optimization is an iterative process that solves the problem repeatedly whereas regularization adds additional information to the problem to find a less accurate but stable solution.

In optimization an objective function $f(x)$, is used to minimize the difference between the actual solution and the desired design

---

* Corresponding author.
E-mail address: kmbryden@iastate.edu (K.M. Bryden).

http://dx.doi.org/10.1016/j.advengsoft.2017.02.009
0965-9978/

[10]. The objective function incorporates all the design variables and is frequently subject to design constraints, which can either be equality constraints *g(x)* or inequality constraints *h(x)*. The search space is the domain of the objective function *f(x),* and computing the value of the objective function at each iteration, given that it satisfies the constraints, is called a feasible solution [44]. Depending on the nature of the desired solution either the design variables or the constraints can be varied until the objective function is sufficiently minimized to find an acceptable solution. This can be expressed as

$$\min f(x) \tag{1}$$

is subject to

$$g(x) = c \tag{2}$$

and

$$h(x) \le d \tag{3}$$

When optimization is used to solve radiant enclosure problems, the objective function takes the form of the equation below. The value of the objective function is found by computing the variance for desired temperature or heat flux profiles on the design surfaces.

$$f = \frac{1}{N}\sum_{i=1}^{N}\left[T_i - T_i^*\right]^2 \tag{4}$$

Where *N* is the number if surfaces, *T* is the current temperature of each design surface and *T** is the desired temperature and *f* is the value of the objective function.

Derivative-based optimization methods move iteratively through the search space, using derivative information to guide the search [14,20]. This procedure is repeated until the stopping criteria have been reached, e.g., a specified number of time steps or the objective function has been minimized below a certain value [32]. These methods include conjugate gradient, Newton-Raphson, golden section, and steepest descent, which have all been used to solve radiant enclosure problems [11,12,19,34]. Heuristic optimization methods have also been used in radiant enclosure problems and include: genetic algorithms, tabu search, simulated annealing and particle swarm optimization [1,36,37]. Heuristic optimization often requires a large amount of sampling of the search space, which means the search may take longer.

Regularization attempts to make the ill-posed portion of the problem tractable at the expense of accuracy. Finding an accurate and stable solution means striking a balance by reducing the fluctuations associated with the ill-posed nature of the problem without producing an over smoothed solution [2,43]. Regularization techniques, such as truncated singular value decomposition, modified truncated singular value decomposition, and Tikhonov regularization have been used to solve radiant enclosure problems [18,27]. Radiant enclosure problems have also been solved using different geometric configurations and initial conditions [13]. Singular value decomposition (SVD) is an algebraic manipulation wherein a matrix of known parameters (A) is broken into three linearly independent matrices, an orthonormal unitary matrix, an orthonormal conjugate transpose matrix, and a diagonal matrix of singular values. The matrix of singular values is used to determine how invertible the matrix is and if the matrix is well posed or ill posed. Truncated singular value decomposition and modified truncated singular value decomposition are based off of singular value decomposition. By truncating some of these singular values, the matrix becomes well posed, and a realizable solution can be found. Modified truncated singular value decomposition adds a correcting term for the remaining singular values and corresponding singular vectors [22]. Tikhonov's regularization procedure attempts to reduce unstable effects by adding smoothing terms to the least squares equation [42]. These methods regularize the system by minimizing the residual and as such are resistant to errors in input data.

In this paper we describe a novel approach to solving a radiant enclosure design problem by posing it as a resource sharing problem which can be solved using distributed construction. Specifically, the distribution and redistribution of a shared resource (temperature) by computational agents allows them to manipulate a shared environment. Unlike optimization, where all of the design variables are used in a single objective function, each agent acts independently and observes a single design variable. Each agent will continue to take action until their local state is met, and when all of the agents meet their local state, the desired global state is met as well.

2. Background

Biological systems have been used as the source of inspiration for many optimization algorithms, e.g., particle swarm optimization, genetic algorithms, and ant colony optimization [45]. Self-organization is an area of study, that draws its origins in biological systems and seeks to establish how organisms can react, adapt, and interact with their environment and each other to create macroscopic level behaviors from microscopic interactions [24]. Flocking behavior of birds and schooling behavior of fish are both examples of how order can spontaneously be created from disorder and how global behaviors are created from local interactions [9].

Stigmergy, like self-organization, uses local information and simple instruction sets to make decisions without global guidance. Stigmergy is how social insects (e.g., bees, ants, termites and wasps) coordinate their behaviors based on indirect communication methods. Coordination is established by making small changes to the insects' environment, which other insects can interpret, and triggering actions or responses, which further alter their environment [6]. Actions reinforce each other and can lead to the construction of complex structures without the need for direct communication between the individuals or a centralized coordinator. Insects use only local information and simple instruction sets to solve complex problems based on emergent behavior where order is created spontaneously and without planning, using insects interactions with their environment. Dorigo's work using pheromones for routing path optimization is one of the seminal works that combines self-organization, insect behavior, and a practical implementation of stigmergy [16]. This current work differs in that it focuses on the stigmergic construction process. For example, papers wasps and African termites and their construction methods have been modeled [5,17,30,40]. This construction process has inspired the development of a number of computational techniques [8,21,35]. Construction of paper wasp nests involves wasps gradually depositing their own building material based only on the current state of the nest [29]. Each wasp evaluates the current state of construction and based on a simple set of rules determines where to build. In the same way African termites build complicated colonies with complex systems for heating, ventilation, cooling, and separate chambers for nurseries and farming [39]. Each termite will perform a task based on their preference, experience, and abilities [7]. Termites operate continuously and efficiently without a centralized coordinator or a task list. Each worker is allowed to contribute and although they use roughly the same rules, slight differences in preferences and abilities leads to emergent behavior.

The computational application of stigmergic construction is *distributed construction*. In distributed construction *Agents* are independent actors, i.e., computational insects. These agents can sense changes to their environment and based on simple rule sets, manipulate uniform construction materials or blocks to change