Under consideration for publication in Theory and Practice of Logic Programming 1
Sol ving Dist ributed Con str aint Optim ization Problems Using Logic Programming
Tie p Le , Tr an Cao Son , En ric o P onte lli , William Ye oh Computer Science Department New Mexico State University Las Cru ces , NM , 8 800 1 , USA E - mail : { tile , t son , ep on tell , w ye oh } @ cs . n ms u . edu
submitted 1 January 2003 ; revised 1 January 2003 ; accepted 1 January 2003
Abstract This paper explores the use of Answer Set Programming (AS P) in solving Dist ributed Con str aint Optim ization Problems (DC OP s). The paper provides the following novel contributions : (1) It shows how one can formulate DC OP s as logic programs ; (2) It introduces ASP - DP OP , the first DC OP algorithm that is based on logic programming ; (3) It experiment ally shows that ASP - DP OP can be up to two orders of magnitude faster than DP OP (its imperative programming counterpart) as well as solve some problems that DP OP fails to solve , due to memory limitations ; and (4) It demonstrates the applic ability of ASP in a wide array of multi - agent problems currently modeled as DC OP s . 1 Under consideration in Theory and Practice of Logic Programming (T PL P).
KEY WOR DS : DC OP ; DP OP ; Logic Programming ; ASP
1 Introduction Dist ributed Con str aint Optim ization Problems (DC OP s) are optimization problems where agents need to coordinate the assignment of values to their local variables to maximize the overall sum of resulting constraint utilities (Mod i et al . 2005 ; Pet cu and F alt ings 2005 a ; Ma iller and Less er 2004 ; Ye oh and Yok oo 2012). The process is subject to limitations on the communication capabilities of the agents ; in particular , each agent can only exchange information with neighboring agents within a given top ology . DC OP s are well - su ited for modeling multi - agent coordination and resource allocation problems , where the primary interactions are between local subs ets of agents . Researchers have used DC OP s to model various problems , such as the distributed scheduling of meetings (Ma hes war an et al . 2004 ; Z ivan et al . 2014), distributed allocation of targets to sensors in a network (Far inelli et al . 2008), distributed allocation of resources in disaster evacuation scenarios (L ass et al . 2008), the distributed management of power distribution networks (K umar et al . 2009 ; J ain et al . 2012), the distributed generation of coalition structures (U eda et al . 2010) and the distributed coordination of logistics operations (Le a ute and F alt ings 2011). 1 This article extends our previous conference paper (Le et al . 2015) in the following manner : (1) It provides a more thorough description of the ASP - DP OP algorithm ; (2) It elabor ates on the algorithm s theoretical properties with complete proofs ; and (3) It includes additional experimental results .

The field has matured considerably over the past decade , since the seminal AD OP T paper (Mod i et al . 2005), as researchers continue to develop more sophisticated solving algorithms . The majority of the DC OP resolution algorithms can be classified in one of three classes : (1) Search - based algorithms , like AD OP T (Mod i et al . 2005) and its variants (Ye oh et al . 2009 ; Ye oh et al . 2010 ; Gutierrez et al . 2011 ; Gutierrez et al . 2013), AFB (G ers h man et al . 2009), and MGM (Ma hes war an et al . 2004), where the agents enumer ate combinations of value assignments in a decentralized manner ; (2) In ference - based algorithms , like DP OP (P etc u and F alt ings 2005 a) and its variants (P etc u and F alt ings 2005 b ; Pet cu and F alt ings 2007 ; Pet cu et al . 2007 ; Pet cu et al . 2008), max - sum (Far inelli et al . 2008), and Action G DL (V iny als et al . 2011), where the agents use dynamic programming techniques to propagate aggreg ated information to other agents ; and (3) Sam pling - based algorithms , like D UCT (Ott ens et al . 2012) and D - G ib bs (N guyen et al . 2013 ; Fi ore tto et al . 2014), where the agents sample the search space in a decentralized manner . The existing algorithms have been designed and developed almost exclusively using imperative programming techniques , where the algorithms define a control flow , that is , a sequence of commands to be executed . In addition , the local sol ver employed by each agent is an ad - h oc implementation . In this paper , we are interested in investigating the benefits of using decl ar ative programming techniques to solve DC OP s , along with the use of a general constraint sol ver , used as a black box , as each agent s local constraint sol ver . Specifically , we propose an integration of Dist ributed Pse udo - tree Optim ization Procedure (DP OP) (P etc u and F alt ings 2005 a), a popular DC OP algorithm , with Answer Set Programming (AS P) (N iem ela 1999 ; Mare k and Tr us z cz yn ski 1999) as the local constraint sol ver of each agent . This paper provides the first step in brid ging the areas of DC OP s and ASP ; in the process , we offer novel contributions to both the DC OP field as well as the ASP field . For the DC OP community , we demonstrate that the use of ASP as a local constraint sol ver provides a number of benefits , including the ability to capitalize on (i) the highly expressive ASP language to more concise ly define input instances (e . g ., by representing constraint utilities as implicit functions instead of explicitly enumer ating their extensions) and (ii) the highly optimized ASP sol vers to exploit problem structure (e . g ., propag ating hard constraints to ensure consistency). For the ASP community , the paper makes the equally important contribution of increasing the applic ability of ASP to model and solve a wide array of multi - agent coordination and resource allocation problems , currently modeled as DC OP s . Furthermore , it also demonstrates that general , off - the - she lf ASP sol vers , which are continuously hon ed and improved , can be coupled with distributed message passing protocols to outper form specialized imperative sol vers . The paper is organized as follows . In Section 2 , we review the basic definitions of DC OP s , the DP OP algorithm , and ASP . In Section 3 , we describe in detail the structure of the novel ASP - based DC OP sol ver , called ASP - DP OP , and its implementation . Section 4 provides an analysis of the properties of ASP - DP OP , including proofs of sound ness and comple teness of ASP - DP OP . Section 5 provides some experimental results , while Section 6 reviews related work . Finally , Section 7 provides conclusions and indications for future work .

2 Background In this section , we present an overview of DC OP s , we describe DP OP , a complete distributed algorithm to solve DC OP s , and provide some fundamental definitions of ASP . 2 . 1 Dist ributed Con str aint Optim ization Problems A Dist ributed Con str aint Optim ization Problem (DC OP) (Mod i et al . 2005 ; Pet cu and F alt ings 2005 a ; Ma iller and Less er 2004 ; Ye oh and Yok oo 2012) can be described as a tuple M = h X , D , F , A , ± i where : X = { x 1 , . . . , x n } is a finite set of (dec ision) variables ; D = { D 1 , . . . , D n } is a set of finite domains , where Di is the domain of the variable x i X , for 1 ī ◘ i ī ◘ n ; F = { f 1 , . . . , f m } is a finite set of constraints , where f j is a k j - ary function f j : Dj 1 Dj 2 . . . Dj kj 7 R { } that specifies the utility of each combination of values of variables in its scope ; the scope is den oted by sc p (f j) = { x j 1 , . . . , x j kj }; 2 A = { a 1 , . . . , ap } is a finite set of agents ; and ± : X 7 A maps each variable to an agent . We say that a variable x is owned by an agent a if ± (x) = a . We denote with ± i the set of all variables that are owned by an agent a i , i . e ., ± i = { x X — ± (x) = a i } . Each constraint in F can be either hard , indicating that some value combinations result in a utility of and must be avoided , or soft , indicating that all value combinations result in a finite utility and need not be avoided . A value assignment is a (partial or complete) function x that maps variables of X to values in D such that , if x (xi) is defined , then x (xi) Di for i = 1 , . . . , n . For the sake of simplicity , and with a slight abuse of notation , we will often denote x (xi) simply with x i . Given a constraint f j and a complete value assignment x for all decision variables , we denote with x f j the projection of x to the variables in sc p (f j); we refer to this as a partial value assignment for f j . For a DC OP M , we denote with
i/s¿