SCLP Semantics for (multi-criteria) Shortest Path Problems

Stefano Bistarelli[1], Ugo Montanari[1], and Francesca Rossi[2]
{bista,ugo,rossi}@di.unipi.it

[1] Dipartimento di Informatica, Università di Pisa, Corso Italia 40, 56125 Pisa.
Tel: +39-050-887268. Fax: +39-050-887226.
[2] Dipartimento di Matematica Pura ed Applicata, Università di Padova, Via Belzoni 7, 35131 Padova.
Tel: +39-049-8275982. Fax: +39-049-8758596.

Abstract. The aim of the paper is to show an example of interaction between the AI and OR fields of research using the Constraint Programming (CP) framework to glue them together. We solve a shortest path problem using the c-semiring-based framework for programming over soft constraints (SCLP), which is an extension of classical constraint solving and programming. The SCLP approac handle uniformly several semirings. Besides the semiring $\langle\{true, false\}, \vee, \cup, false, true\rangle$, which is the ordinary one for constraints, it handles the semiring $\langle\mathcal{R}^+, \min, +, +\infty, 0\rangle$, which represents minimization. Moreover, the cartesian product of two c-semirings is still a c-semiring. Using SCLP, a variety of multi-criteria optimization can be represented and solved in a uniform way, and heuristics and local search algorithms already used in the OR community for a totally ordered criterion, may be studied and extended to partial orders of (one or more) criteria.

1 Motivations

Shortest Path (SP) problems [4] are among the most studied network optimization problems. They are mainly used to represent and solve transportation problems, where the optimization may involve different criteria, says cost, time, resources, etc. Most interesting is the multi-criteria case, where the optimization involves a set of criteria and the solution algorithms are extensions of the ordinary ones [4].

In this paper we consider multi-criteria SP problems and we suggest to model them via constraint logic programs (CLP) over soft constraints. The basic idea is that soft constraints allow to faithfully represent the optimization criteria, and CLP provides a declarative way to describe the SP problems. Moreover, this way of modelling and solving SP problems allows to associate to such problems both a declarative and an operational semantics.

The paper is organized as follows: Section 2 gives the basic notions of soft constraints (using the semiring-based framework) and soft CLP. Then, Section 3 shows the construction to pass from an SP problem to a CLP program, and finally Section 4 concludes the paper by summarizing its contributions.

2 Soft CLP

In this paper we will use the semiring-based CLP framework (SCLP) [3] to describe SP problems in a declarative fashion. The framework is based on the notion of *c-semiring* introduced in [1, 2]. A c-semiring S is a tuple $\langle A, +, \times, \mathbf{0}, \mathbf{1}\rangle$ where A is a set with two special elements ($\mathbf{0}, \mathbf{1} \in A$) and with two operations $+$ and $\times$ that satisfy certain properties: $+$ is closed, commutative, associative, idempotent, $\mathbf{0}$ is its unit element and $\mathbf{1}$ is its absorbing element; $\times$ is closed, associative, commutative, distributes over $+$, $\mathbf{1}$ is its unit element, and $\mathbf{0}$ is its absorbing element.

The $+$ operation defines a partial order $\leq_S$ over A such that $a \leq_S b$ iff $a + b = b$; we say $a \leq_S b$ if b represents a value *better* than a. Other properties related to the two operations are: $+$ and $*$ are monotone on $\leq_S$; $\mathbf{0}$ is its minimum and $\mathbf{1}$ its maximum; $\langle A, \leq_S\rangle$ is a complete lattice and $+$ is its lub. Finally, if $\times$ is idempotent then: $+$ distribute over $\times$; $\langle A, \leq_S\rangle$ is a complete distributive lattice and $\times$ its glb.

Semiring-based Constraint Satisfaction Problems (SCSPs) are constraint problems where each variable instantiation is associated to an element of A (to be interpreted as a cost, level of preference, ...), and constraints are combined via the $\times$ operation and compared via the $\leq_S$ ordering. Varying the set A and the meaning of the $+$ and $\times$ operations, we can represent fuzziness (using the semiring $S_{FCSP} = \langle\{x \mid x \in [0,1]\}, max, min, 0, 1\rangle$), partially know problem (using a probabilistic semiring $S_{prob} = \langle\{x \mid x \in [0,1]\}, max, \times, 0, 1\rangle$), and optimization ($S_{WCSP} = \langle\mathcal{R}^+, \min, +, +\infty, 0\rangle$). Moreover the cartesian product of two c-semirings is another c-semiring, and this can be fruitfully used to describe multicriteria constraint satisfaction and optimization problems.

Figure 1 shows an example of a fuzzy constraint problem, where variables are circles, constraints are arcs, and semiring values are written to the right of the corresponding tuples. The solutions of this SCSP are pair (that is, an assignement for each of the two variables) and their S_{FCSP} semiring value is obtained by looking at the value obtained combining using the $\times$ operation, that is min, all the subtuples (as many as the constraints) forming the pair. For example, the pair of domain assignement $\langle a, b\rangle$ take the value $min\{0.9, 0.2, 0.5\} = 0.2$. In some cases we are interested to know the value of the assignement of only one variable, say $X = a$. In this case we have to project all the pair involving the variable X with value a using the $+$ operator that is the max. The assignement $X = a$ take max of the assignement of the pair $\langle a, b\rangle$ and $\langle a, a\rangle$ that is $max\{0.2, 0.8\} = 0.2$.

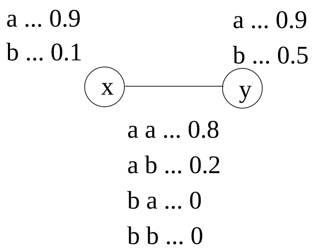

Fig. 1. A fuzzy CSP.

The Semiring-based CLP framework (SCLP) extends the CLP formalism in order to be able to handle also SCSP [1, 2] problems. In passing from CLP to SCLP languages, we had to replace classical constraints with the more general SCSP constraints where we are able to assign a *level of preference* to each instantiated constraint (i.e. a ground atom). To do this, we also had to modify the notions of interpretation, model, model intersection, and others, since we have to take into account the semiring operations and not the usual CLP operations. For example, while CLP interpretations associate a truth value (either *true* or *false*) to each ground atom, here ground atoms must be given one of the elements of the semiring. Also, while in CLP the value associated to an existentially quantified atom is the *logical or* among the truth values associated to each of its instantiations, here we had to replace the *or* with another operation which refers to one of the semiring operations (the $+$). The combination of atoms, which in CLP is modeleled via *logical and*, in SCLP is instead handled via the $\times$ operation of the semiring.

Besides the model-theoretic semantics based on models and interpretations, SCLP programs come also with a fixpoint and an operational semantics. These semantics are conservative extensions of the corresponding ones for logic programming (LP), since by choosing a particular semiring (the one with just two elements, *true* and *false*, and the logical *and* and *or* as the two semiring operations) we get exactly the LP semantics. The presence of a possibly partial order among the elements of the semiring makes the operational semantics more complex to compute. In fact, there could be several refutations for a goal which lead to different semiring elements which are not comparable in the partial order. In this case, these elements have to be combined using in this case $+$ in order to get the solution corresponding to the given goal, and their combination could not be reachable by any derivation path in the search tree.

The fact that we have to keep track of several refutation paths when we have a partial order (instead of a total one) can be fruitfully used in the context of this paper when we have an SP problem with incomparable costs in the arcs. In fact, in the case of a partial order, the solution of an SP problem should consists of all those paths which are not "dominated" by others.

The three semantics are shown to be equivalent. In particular, we prove that, given the set of all refutations starting from a given goal, it is possible to derive the declarative meaning of both the existential closure of the goal and its universal closure.

A simple example of an SCLP program over the semiring $\langle -N \cup \{-\infty\}, max, +, -\infty, 0\rangle$, where $-N$ is the set of non-positive integers and $D = \{a, b, c\}$, is the following one:

```
s(x)   :- p(x,y).
p(a,b) :- q(a).
p(a,c) :- r(a).
q(a)   :- t(a).
t(a)   :- -2.
r(a)   :- -3.
```

The choice of this semiring allows to represent constraint optimization problems where the semiring elements are the costs for the instantiated atoms. Note that the ordering $\leq_S$ in this semiring coincides with the $\leq$ ordering over integers. The intuitive meaning of a semiring value like -3 associated to the atom $r(a)$ is that $r(a)$ costs 3 units. Thus the set $-N \cup \{-\infty\}$ contains all possible costs, and the choice of the two operations max and $+$ implies that we intend to maximize the sum of the costs. This gives us the possibility to select the atom instantiation which gives the minimal cost overall. For example the value assigned to $p(a, c)$ is the same assigned to $r(a)$, which take the semiring value -3. For a more complex example consider the goal $s(a)$; in this case we assign the semiring value represented by the max of the value assigned at the atom $p(a, b)$ and $p(a, c)$ which is $max\{-2, -3\} = -2$.

3 From SP problems to SCLP programs

An SP problem can be represented by a graph $G = (N, E)$, where each arc $pq \in E$ has associated a label representing the cost of the arc from node p to node q. Given two nodes, a solution of the problem consists of finding a path (or a set of paths) between the two nodes, whose cost is minimal.

To describe this classical problem, we consider an SCLP program over the semiring $S = \langle \mathcal{R}^+, \min, +, +\infty, 0\rangle$, which is an appropriated framework to represent constraint problems where one wants to minimize the costs of the path solutions[1]. Note that a cost of a path is computed by combining the cost of each arc using the $\times$ operator of the semiring (that is $+$).

Here we will consider SP problems where we are interested in optimizing more than one criteria. Consider for example the SP problem represented in Figure 2; each arc has associated a pair representing the weight of the arc in terms of *cost* and *time*. Given any node, we want to find a path from this node to v (if it exists) that minimizes both the criteria [2].

We can translate it into the following SCLP program over the semiring

$$\langle (\mathcal{R}^+)^2, \text{min'}, +', \langle +\infty, +\infty\rangle, \langle 0, 0\rangle\rangle,$$

where min' and $+'$ are classical min and $+$, suitably extended to pairs.

```
p(a) :- pq, q(x).
p(b) :- pr, r(x).
q(a) :- qs, s(x).
```

[1] Note that this semiring is an alternative characterization of optimization problems with respect to the semiring used in the example of SCLP in the previous section.

[2] Note that we also label arcs with letters $a, b, \ldots$ to keep track of the arcs used in a certain path.

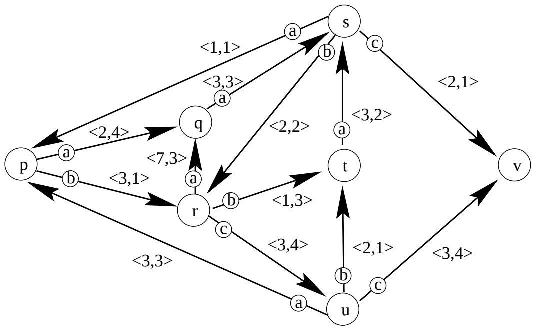

Fig. 2. A multicriteria SP problem.

```
r(a) :- rq, q(x).
r(b) :- rt, t(x).
r(c) :- ru, u(x).
s(a) :- sp, p(x).
s(b) :- sr, r(x).
s(c) :- sv, v(x).
t(a) :- ts, s(x).
u(a) :- up, p(x).
u(b) :- ut, t(x).
u(c) :- uv, v(x).
v(x) :- <0,0>.
pq   :- <2,4>.
pr   :- <3,1>.
qs   :- <3,3>.
rq   :- <7,3>.
rt   :- <1,3>.
ru   :- <3,4>.
sp   :- <1,1.
sr   :- <2,2>.
sv   :- <2,1>.
ts   :- <3,2>.
up   :- <3,3>.
ut   :- <2,1>.
uv   :- <3,4>.
```

In words, each arc in the graph is mapped onto a corresponding atom, which is then associated to its weight, consisting of a cost and a duration, via a clause. For example, the arc from p to q, which has cost 2 and duration 4, is represented by the clause `pq :- <2,4>`. Moreover, each node is mapped onto a predicate with one argument, whose instantiations represent all possible ways to move to another node in the graph. For each of such moves, we have a clause whose body specifies the arc to be used in such a move and the destination node. For example, from node p we can go to node q via the arc labelled a. Thus we have the clause `p(a) :- pq, q(x)`. Note that the argument of predicate q is free, since from q we can then move choosing any outgoing arc.

To compute the solution of the problem it is enough to perform a query in the SCLP framework; for example, if we want to compute the cost and time of the path from r to v we have to perform the query $\exists x.\ r(x)$. We obtain the value $\langle 6,4\rangle$ that represents a lower bound of the weights of all the paths from r to v. Moreover, since the instantiated predicate r produces the output $\{r(a) = \langle 12,7\rangle, r(b) = \langle 6,4\rangle, r(c) = \langle 6,8\rangle\}$, we can conclude that the best path from r to v uses the arc rt. To compute the complete path solution we have to perform the same query also for t and s.

Note that we obtain a unique path as solution since there exists one path from r to v which is better than all others. If we consider instead the node p we obtain for $\exists x.\ p(x)$ a value $\langle 7,5\rangle$. This value represent a lower bound of the costs of all the paths, but do not represent a specific path; in fact, the values of the instantiated predicate p are: $p(a) = \langle 7,8\rangle$ and $p(b) = \langle 9,5\rangle$. In this case the best path is represented as a tree (or a set of paths).

4 Conclusions

We have shown with a simple example how to model a shortest path problem as a CLP program over soft constraints. This automatically provides this kind of problems with not only an operational semantics but also a declarative one. Our modelization allows for a faithful representation of multi-criteria scenarios. Moreover, our framework does not force one to combine the different criteria in a *generalized* one as done in [5], but is instead able to handle all the criteria together, in a semiring which is able to capture the peculiarities of the problems under consideration. At our knowledge, no general framework to capture and describe uniformly this type of problems has been defined until now.

Although this study is very preliminary, we think it is promising, in that one can already see, in prespective, the following advantages:

- multi-criteria SP problems can be embedded in our framework and modified versions of classical algorithms used for SP problems can be possibly used to solve them. Candidates are a version of the Dijkstra algorithm [6] and the gaussian elimination method, which have to be modified to take into account the partial (and not total) order induced by the semiring.
- The algorithms used to find the shortest path could be used to provide an efficient way to give an operational semantics to SCLP programs. In fact, in general, the operational semantics computes the *best* value associated to a predicate by looking for all the best *simple* refutation trees of an SCLP program. By using an efficient way to find, and possibly cut away, the *dominated* paths, the applicability of the framework could be improved.

References

1. S. Bistarelli, U. Montanari and F. Rossi. Constraint Solving over Semirings. In *Proceedings of IJCAI'95*, Morgan Kaufman, 1995.
2. S. Bistarelli, U. Montanari and F. Rossi. Semiring-based Constraint Solving and Optimization. *Journal of ACM*, vol. 44, no. 2, March 1997.
3. S. Bistarelli, U. Montanari, and F. Rossi. Semiring-based Constraint Logic Programming. In *Proc. IJCAI97*. Morgan Kaufman, 1997.
4. S. Pallottino e M. G. Scutellà. Shortest path algorithms in transportation models: classical and innovative aspects. In (P. Marcotte and S. Nguyen, eds.) Equilibrium and Advanced Transportation Modelling, Kluwer (1998) 245-281.
5. Y. Israeli and A. Ceder. In Multiobjective approach for designing transit routes with frequencies. *Advanced methods in transportation analysis*, L. Bianco and P. Toth, eds., Springer, Berlin, 1996, 157-182.
6. E. Dijkstra. A note on two problems in connection with graphs. Numer. Math. 1 (1959), 269-271.