Identifying Cost-Effective Common Subexpressions to Reduce Operation Count in Tensor Contraction Evaluations

Albert Hartono[1], Qingda Lu[1], Xiaoyang Gao[1], Sriram Krishnamoorthy[1], Marcel Nooijen[3], Gerald Baumgartner[4], David E. Bernholdt[6], Venkatesh Choppella[1,7], Russell M. Pitzer[2], J. Ramanujam[5], Atanas Rountev[1], and P. Sadayappan[1]

[1] Dept. of Computer Science and Engineering
[2] Dept. of Chemistry
The Ohio State University, Columbus, OH 43210, USA
[3] Dept. of Chemistry, University of Waterloo, Waterloo, Ontario N2L BG1, Canada
[4] Dept. of Computer Science
[5] Dept. of Electrical and Computer Engineering
Louisiana State University, Baton Rouge, LA 70803, USA
[6] Computer Sci. & Math. Div., Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA
[7] Indian Institute of Information Technology and Management, Kerala, India

Abstract. Complex tensor contraction expressions arise in accurate electronic structure models in quantum chemistry, such as the coupled cluster method. Transformations using algebraic properties of commutativity and associativity can be used to significantly decrease the number of arithmetic operations required for evaluation of these expressions. Operation minimization is an important optimization step for the Tensor Contraction Engine, a tool being developed for the automatic transformation of high-level tensor contraction expressions into efficient programs. The identification of common subexpressions among a set of tensor contraction expressions can result in a reduction of the total number of operations required to evaluate the tensor contractions. In this paper, we develop an effective algorithm for common subexpression identification and demonstrate its effectiveness on tensor contraction expressions for coupled cluster equations.

1 Introduction

Users of current and emerging high-performance parallel computers face major challenges to both performance and productivity in the development of their scientific applications. For example, the manual development of accurate quantum chemistry models typically takes an expert several months of tedious effort; high-performance implementations can take substantially longer. One approach to address this situation is the use of automatic code generation to synthesize efficient parallel programs from the equations to be implemented, expressed in a very high-level domain-specific language. The Tensor Contraction Engine (TCE) [3, 2] is such a tool, being developed through a collaboration between computer scientists and quantum chemists.

The first step in the TCE's code synthesis process is the transformation of input equations into an equivalent form with minimal operation count. Equations typically range from around ten to over a hundred terms, each involving the contraction of two or more tensors, and most quantum chemical methods involve two or more coupled equations of this type. This optimization problem can be viewed as a generalization of the matrix chain multiplication problem, which, unlike the matrix-chain case, has been shown to be NP-hard [6]. Our prior work focused on the use of single-term optimization (strength reduction or parenthesization), which decomposes multi-tensor contraction operations into a sequence of binary contractions, coupled with a global search of the composite single-term solution space for factorization opportunities. Exhaustive search (for small cases) and a number of heuristics were shown to be effective in minimizing the operation count [4].

Common subexpression elimination (CSE) is a classical optimization technique used in traditional optimizing compilers [1] to reduce the number of operations, where intermediates are identified that can be computed once and stored for use multiple times later. CSE is routinely used in the manual formulation of quantum chemical methods, but because of the complexity of the equations, it is extremely difficult to explore all possible formulations manually. CSE is a powerful technique that allows the exploration of the much larger algorithmic space than our previous approaches to operation minimization. However, the cost of the search itself grows explosively. In this paper, we develop an approach to CSE identification in the context of operation minimization for tensor contraction expressions. The developed approach is shown to be very effective, in that it automatically finds efficient computational forms for challenging tensor equations.

Quantum chemists have proposed domain-specific heuristics for strength reduction and factorization for specific forms of tensor contraction expressions (e.g., [7, 9]). However, their work does not consider the general form of arbitrary tensor contraction expressions. Single-term optimizations in the context of a general class of tensor contraction expressions were addressed in [6]. Approaches to single-term optimizations and factorization of tensor contraction expressions were presented in [4, 8]. Common subexpression identification to enhance single-term optimization was not considered in any of these approaches.

The rest of this paper is organized as follows. Section 2 provides a more detailed description of the operation minimization and the common subexpression elimination problem in the context of tensor contraction expressions. Section 3 describes our approach. Experimental results are presented in Section 4 and Section 5 concludes the paper.

2 Common Subexpressions and Operation Count Reduction

A tensor contraction expression comprises a sum of a number of terms, where each term might involve the contraction of two or more tensors. We first illustrate the issue of operation minimization for a single term, before addressing the issue of finding common subexpressions to optimize across multiple terms. Consider the following tensor contraction expression involving three tensors t, f and s, with indices x and z that have range V, and indices i and k that have range O. Distinct ranges for different indices is a characteristic of the quantum chemical methods of interest, where O and V correspond to the number of occupied and virtual orbitals in the representation of the molecule (typically $V \gg O$). Computed as a single nested loop computation, the number of arithmetic operations needed would be $2O^2V^2$.

$$r_i^x = \sum_{z,k} t_i^z f_z^k s_k^x \qquad (\text{cost}=2O^2V^2)$$

However, by performing a two-step computation with an intermediate I, it is possible to compute the result using $4OV^2$ operations:

$I_z^x = \sum_k f_z^k s_k^x$ (cost=$2OV^2$); $r_i^x = \sum_z t_i^z I_z^x$ (cost=$2OV^2$)

Another possibility using $4O^2V$ computations, which is more efficient when $V > O$ (as is usually the case in quantum chemistry calculations), is shown below:

$I_i^k = \sum_z t_i^z f_z^k$ (cost=$2O^2V$); $r_i^x = \sum_k I_i^k s_k^x$ (cost=$2O^2V$)

The above example illustrates the problem of single-term optimization, also called strength reduction: find the best sequence of two-tensor contractions to achieve a multi-tensor contraction. Different orders of contraction can result in very different operation costs; for the above example, if the ratio of V/O were 10, there is an order of magnitude difference in the number of arithmetic operations for the two choices.

With complex tensor contraction expressions involving a large number of terms, if multiple occurrences of the same subexpression can be identified, it will only be necessary to compute it once and use it multiple times. Thus, common subexpressions can be stored as intermediate results that are used more than once in the overall computation. Manual formulations of computational chemistry models often involve the use of such intermediates. The class of quantum chemical methods of interest, which include the coupled cluster singles and doubles (CCSD) method [7, 9], are most commonly formulated using the molecular orbital basis (MO) integral tensors. However the MO integrals are intermediates, derived from the more fundamental atomic orbital basis (AO) integral tensors. Alternate "AO-based" formulations of CCSD have been developed in which the more fundamental AO integrals are used directly, without fully forming the MO integrals [5]. However it is very difficult to manually explore all possible formulations of this type to find the one with minimal operation count, especially since it can depend strongly on the characteristics of the particular molecule being studied.

The challenge in identifying cost-effective common subexpressions is the combinatorial explosion of the search space, since single-term optimization of different product terms must be treated in a coupled manner. The following simple example illustrates the problem.

Suppose we have two MO-basis tensors, v and w, which can be expressed as a transformation of the AO-basis tensor, a, in two steps. Using single-term optimization to form tensor v, we consider two possible sequences of binary contractions as shown below, which both have the same (minimal) operation cost. Extending the notation above, indices p and q represent AO indices, which have range $M = O + V$.

Seq. 1: $I1_q^i = \sum_p a_q^p c_p^i$ (cost=$2OM^2$); $v_j^i = \sum_p I1_p^i d_j^p$ (cost=$2O^2M$)

Seq. 2: $I2_i^p = \sum_q a_q^p d_i^q$ (cost=$2OM^2$); $v_j^i = \sum_p I2_j^p c_p^i$ (cost=$2O^2M$)

To generate tensor w, suppose that there is only one cost-optimal sequence:

$I1_q^i = \sum_p a_q^p c_p^i$ (cost=$2OM^2$); $w_x^i = \sum_p I1_p^i e_x^p$ (cost=$2OVM$)

Note that the first step in the formation of w uses the same intermediate tensor $I1$ that appears in sequence 1 for v. Considering just the formation of v, either of the two sequences is equivalent in cost. But one form uses a common subexpression that is useful in computing the second MO-basis tensor, while the other form does not. If sequence 1 is chosen for v, the total cost of computing both v and w is $2OM^2 + 2O^2M + 2OVM$. On the other hand, the total cost is higher if sequence 2 is chosen ($4OM^2 + 2O^2M + 2OVM$). The $2OM^2$ cost difference is significant when M is large.

When a large number of terms exist in a tensor contraction expression, there is a combinatorial explosion in the search space if all possible equivalent-cost forms for each product term must be compared with each other.

In this paper, we address the following question: By developing an automatic operation minimization procedure that is effective in identifying suitable common subexpressions in tensor contraction expressions, can we automatically find more efficient computational forms? For example, with the coupled cluster equations, can we automatically find AO-based forms by simply executing the operation minimization procedure on the standard MO-based CCSD equations, where occurrences of the MO integral terms are explicitly expanded out in terms of AO integrals and integral transformations?

3 Algorithms for Operation Minimization with CSE

In this section, we describe the algorithm used to perform operation minimization, by employing single-term optimization together with CSE. The exponentially large space of possible single-term optimizations, together with CSE, makes an exhaustive search approach prohibitively expensive. So we use a two-step approach to apply single-term optimization and CSE in tandem.

The algorithm is shown in Fig. 2. It uses the single-term optimization algorithm, which is broadly illustrated in Fig. 1 and described in greater detail in our earlier work [4]. It takes as input a sequence of tensor contraction statements. Each statement defines a tensor in terms of a sum of tensor contraction expressions. The output is an optimized sequence of tensor contraction statements involving only binary tensor contractions. All intermediate tensors are explicitly defined.

The key idea is to determine the parenthesization of more expensive terms before the less expensive terms. The most expensive terms contribute heavily to the overall operation cost, and potentially contain expensive subexpressions. Early identification of these expensive subexpressions can facilitate their reuse in the computation of other expressions, reducing the overall operation count.

The algorithm begins with the *term set* to be optimized as the set of all the terms of the tensor contraction expressions on the right hand side of each statement. The set of intermediates is initially empty. In each step of the iterative procedure, the parenthesization for one term is determined. Single-term optimization is applied to each term in the term set using the current set of intermediates and the most expensive term is chosen to be parenthesized. Among the set of optimal parenthesizations for the chosen term, the one that maximally reduces the cost of the remaining terms is chosen. Once the term and its parenthesization are decided upon, the set of intermediates is updated and the corresponding statements for the new intermediates are generated. The procedure continues until the term set is empty.

4 Experimental Results

We evaluated our approach by comparing the optimized operation count of the MO-based CCSD T1 and T2 computations with the corresponding equations in which the occurrences of MO integrals are replaced by the expressions that produce them, referred to as the expanded form. Table 1 illustrates the characteristics of CCSD T1 and T2 equations. Fig. 3 shows the CCSD T1 equation, consisting of the computation of the

```
SINGLE-TERM-OPT-CSE($E, is$)
1  if $E$ is a single-tensor expression
2    then return $\{\langle E, \emptyset \rangle\}$
3    else  \* $E$ is a multiple-tensor contraction expression (i.e., $E_1 * \ldots * E_n$) * \
4          $\{\langle p_1, is_1 \rangle, \langle p_2, is_2 \rangle, \ldots\} \leftarrow$
5              set of pairs of optimal parenthesization of $E$ and its corresponding intermediate set
6              (the given intermediate set $is$ is used to find effective common subexpressions)
7          return $\{\langle p_1, is_1 \rangle, \langle p_2, is_2 \rangle, \ldots\}$
```

Fig. 1. Single-term optimization algorithm with common subexpression elimination

```
OPTIMIZE($stmts$)
 1  $MSET \leftarrow$ set of all terms obtained from RHS expressions of $stmts$
 2  $is \leftarrow \emptyset$ \* the set of intermediates * \
 3  while $MSET \neq \emptyset$
 4  do $M_{heaviest} \leftarrow$ the heaviest term in $MSET$
 5            (searched by applying SINGLE-TERM-OPT-CSE($M_i, is$) on each term $M_i \in MSET$)
 6      $PSET \leftarrow$ SINGLE-TERM-OPT-CSE($M_{heaviest}, is$)
 7      $\langle p_{best}, is_{best} \rangle \leftarrow$ NIL
 8      $profit \leftarrow 0$
 9      for each $\langle p_i, is_i \rangle \in PSET$
10      do $cur\_profit \leftarrow 0$
11          for each $M_i \in (MSET - \{M_{heaviest}\})$
12          do $base\_cost \leftarrow$ op-cost of optimal parenth. in SINGLE-TERM-OPT-CSE($M_i, is$)
13              $opt\_cost \leftarrow$ op-cost of optimal parenth. in SINGLE-TERM-OPT-CSE($M_i, is \cup is_i$)
14              $cur\_profit \leftarrow cur\_profit + (base\_cost - opt\_cost)$
15          if $(\langle p_{best}, is_{best} \rangle = \text{NIL}) \vee (cur\_profit > profit)$
16            then $\langle p_{best}, is_{best} \rangle \leftarrow \langle p_i, is_i \rangle$
17                 $profit \leftarrow cur\_profit$
18      $stmts \leftarrow$ replace the term $M_{heaviest}$ in $stmts$ with $p_{best}$
19      $MSET \leftarrow MSET - \{M_{heaviest}\}$
20      $is \leftarrow is \cup is_{best}$
21  return $stmts$
```

Fig. 2. Global operation minimization algorithm

MO integrals (Steps 1a–1d) and the expression for the single-excitation residual (Step 2). Whereas our examples above used rank-2 tensors for simplicity, the CCSD equations primarily involve rank-4 integral tensors.

The number of arithmetic operations depends upon O and V, which are specific to the molecule and quality of the simulation, but a typical range is $1 \leq V/O \leq 100$. To provide concrete comparisons, we set O to 10 and V to 100 or 500.

The CCSD computation proceeds through a number of iterations in which the AO integrals remain unchanged. At convergence, the amplitudes t_vo and t_vvoo attain values such that the residual vector in Step 2 of Fig. 3 is equal to zero and this typically takes 10–50 iterations. In different variants of CCSD, the MO integrals may also remain unchanged, or may change at each iteration, requiring the AO-to-MO transformation to be repeated. To represent these two cases, we use iteration counts of 10 and 1, respectively, to evaluate the different formulations obtained.

Table 1. Characteristics of input equations used in experiments

Equation	Number of terms	MO Integrals
CCSD T1	14	$v\_ooov, v\_oovv, v\_ovov, v\_ovvv$
CCSD T2	31	$v\_oooo, v\_ooov, v\_oovv, v\_ovoo, v\_ovov, v\_ovvv, v\_vvoo, v\_vvov, v\_vvvv$

$$\text{(1a)}\quad v\_ooov^{h1h2}_{h3p1} = (c\_mo^{q1}_{h3} * c\_mv^{q2}_{p1} * c\_om^{h1}_{q3} * c\_om^{h2}_{q4} * a\_mmmm^{q3q4}_{q1q2})$$

$$\text{(1b)}\quad v\_oovv^{h1h2}_{p1p2} = (c\_mv^{q1}_{p1} * c\_mv^{q2}_{p2} * c\_om^{h1}_{q3} * c\_om^{h2}_{q4} * a\_mmmm^{q3q4}_{q1q2})$$

$$\text{(1c)}\quad v\_ovov^{h1p1}_{h2p2} = (c\_mo^{q1}_{h2} * c\_mv^{q2}_{p2} * c\_om^{h1}_{q3} * c\_vm^{p1}_{q4} * a\_mmmm^{q3q4}_{q1q2})$$

$$\text{(1d)}\quad v\_ovvv^{h1p1}_{p2p3} = (c\_mv^{q1}_{p2} * c\_mv^{q2}_{p3} * c\_om^{h1}_{q3} * c\_vm^{p1}_{q4} * a\_mmmm^{q3q4}_{q1q2})$$

$$\begin{aligned}
\text{(2)}\quad \text{residual}^{p2}_{h1} = {} & 0.25 * (t\_vvoo^{p2p1}_{h2h1} * f\_ov^{h2}_{p2}) - 0.25 * (v\_ovov^{h2p1}_{h1p2} * t\_vo^{p2}_{h2}) \\
& + 0.25 * (f\_vv^{p1}_{p2} * t\_vo^{p2}_{h1}) - 0.25 * (f\_oo^{h2}_{h1} * t\_vo^{p1}_{h2}) + 0.25 * f\_vo^{p1}_{h1} \\
& - 0.25 * (t\_vo^{p1}_{h2} * t\_vo^{p2}_{h1} * t\_vo^{p3}_{h3} * v\_oovv^{h2h3}_{p2p3}) \\
& + 0.25 * (t\_vvoo^{p2p1}_{h2h1} * t\_vo^{p3}_{h3} * v\_oovv^{h2h3}_{p2p3}) - 0.125 * (t\_vo^{p1}_{h2} * t\_vvoo^{p2p3}_{h3h1} * v\_oovv^{h3h2}_{p2p3}) \\
& - 0.125 * (t\_vo^{p2}_{h1} * t\_vvoo^{p3p1}_{h2h3} * v\_oovv^{h2h3}_{p3p2}) - 0.25 * (t\_vo^{p2}_{h1} * v\_ovvv^{h2p1}_{p2p3} * t\_vo^{p3}_{h2}) \\
& - 0.25 * (t\_vo^{p1}_{h2} * v\_ooov^{h2h3}_{h1p2} * t\_vo^{p2}_{h3}) - 0.25 * (t\_vo^{p1}_{h2} * t\_vo^{p2}_{h1} * f\_ov^{h2}_{p2}) \\
& + 0.125 * (t\_vvoo^{p2p3}_{h2h1} * v\_ovvv^{h2p1}_{p2p3}) + 0.125 * (t\_vvoo^{p2p1}_{h2h3} * v\_ooov^{h2h3}_{h1p2})
\end{aligned}$$

Fig. 3. The input formulation of CCSD T1. For compactness, summations are implicit wherever the same index appears twice in a term.

Tables 2 and 3 illustrate the results obtained by optimizing CCSD T1 and T2 equations with the algorithm described above. The total operation counts are shown for different (O,V) pairs, changing iteration counts, and choice of MO integrals to expand. We applied single-term optimization and CSE to the AO-to-MO calculation and the MO-basis expression separately, without expanding any MO integrals - this is representative of current implementations of coupled cluster methods. We report the operation count reduction using our approach relative to the optimized conventional two-step formulation as discussed above.

Among all the sixteen cases we have studied, twelve of them yield a reduction factor ranging from 2.14 to 14.75 and two of them have a reduction factor close to 1.2. We can conclude that our algorithm performs well in practice in most cases. The following observations can be made from the results in Tables 2 and 3.

– The benefits decrease with an increase of the iteration count;
– The benefits increase with increasing number of explicitly expanded terms; and
– The benefits are greater when the V/O ratio is large.

Fig. 4 shows an optimized formulation of the CCSD T1 equation in Fig. 3, when $(O,V) = (10,500)$ and the MO integrals v_ovvv, v_ooov, v_ovov are expanded. It may be seen that this form, with an operation-count reduction factor of 2.49, is significantly different from the original MO-basis formulation in Fig. 3. In this new formulation, the *it* arrays are the common subexpressions identified to reduce the operation count.

5 Conclusions

In this paper, we presented a coupled approach of utilizing single-term optimization and identification of common subexpressions to reduce the operation count in the evaluation of tensor contraction expressions. The benefits of the approach were shown by

Table 2. Results of optimizing CCSD T1 with our algorithm

(O,V)	Iteration Count	Expanded Tensors	Total Operation Count	Reduction Factor
$(10,100)$	1	*None*	1.12×10^{10}	1
		v_ovvv	5.25×10^{9}	2.14
		v_ovvv, v_ooov, v_ovov	4.52×10^{9}	2.48
	10	*None*	1.40×10^{10}	1
		v_ovvv	1.20×10^{10}	1.17
		v_ovvv, v_ooov, v_ovov	1.18×10^{10}	1.19
$(10,500)$	1	*None*	5.36×10^{12}	1
		v_ovvv	1.59×10^{12}	3.37
		v_ovvv, v_ooov, v_ovov	1.51×10^{12}	3.55
	10	*None*	5.63×10^{12}	1
		v_ovvv	2.34×10^{12}	2.41
		v_ovvv, v_ooov, v_ovov	2.26×10^{12}	2.49

Table 3. Results of optimizing CCSD T2 with our algorithm

(O,V)	Iteration Count	Expanded Tensors	Total Operation Count	Reduction Factor
(10,100)	1	*None*	1.51×10^{11}	1
		v_vvvv	6.87×10^{10}	2.20
		v_vvvv, v_ovvv, v_vvov	5.40×10^{10}	2.80
	10	*None*	4.68×10^{11}	1
		v_vvvv	4.68×10^{11}	1
		v_vvvv, v_ovvv, v_vvov	4.67×10^{11}	1
(10,500)	1	*None*	2.85×10^{14}	1
		v_vvvv	2.72×10^{13}	10.48
		v_vvvv, v_ovvv, v_vvov	1.93×10^{13}	14.75
	10	*None*	4.22×10^{14}	1
		v_vvvv	1.76×10^{14}	2.40
		v_vvvv, v_ovvv, v_vvov	1.67×10^{14}	2.53

expanding the tensor contraction expressions in two representative computations, and demonstrating a reduction in the operation count for the composite computation.

Acknowledgments This work has been supported in part by: the U.S. National Science Foundation through grants 0121676, 0121706, 0403342, 0508245, 0509442, 0509467, and 0541409; the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory (ORNL); and a Discovery grant 262942-03 from the Natural Sciences and Engineering Research Council of Canada. ORNL is managed by UT-Battelle, LLC, for the US Dept. of Energy under contract DE-AC-05-00OR22725.

References

1. A. Aho, R. Sethi, and J. Ullman. *Compilers: Principles, Techniques, and Tools*. Addison-Wesley, 1986.
2. A. Auer, G. Baumgartner, D. Bernholdt, A. Bibireata, V. Choppella, D. Cociorva, X. Gao, R. Harrison, S. Krishanmoorthy, S. Krishnan, C. Lam, M. Nooijen, R. Pitzer, J. Ramanujam, P. Sadayappan, and A. Sibiryakov. Automatic code generation for many-body electronic

$$\text{(1a)}\quad it\_1^{q1h1}_{q2q3} = (a\_mmmm^{q4q1}_{q2q3} * c\_om^{h1}_{q4})$$
$$\text{(1b)}\quad it\_2^{h1h2}_{p1q1} = (c\_mv^{q2}_{p1} * (c\_om^{h1}_{q3} * it\_1^{q3h2}_{q1q2}))$$
$$\text{(1c)}\quad v\_oovv^{h1h2}_{p1p2} = (c\_mv^{q1}_{p1} * it\_2^{h2h1}_{p2q1})$$
$$\text{(1d)}\quad it\_3^{q1}_{h1} = (c\_mv^{q1}_{p1} * t\_vo^{p1}_{h1})$$
$$\text{(1e)}\quad it\_4^{h1h2}_{h3p1} = (c\_mo^{q1}_{h3} * it\_2^{h1h2}_{p1q1})$$
$$\text{(1f)}\quad it\_5^{q1}_{q2} = (it\_1^{q1h1}_{q2q3} * it\_3^{q3}_{h1})$$
$$\text{(1g)}\quad it\_6^{h1}_{p1} = (v\_oovv^{h1h2}_{p1p2} * t\_vo^{p2}_{h2})$$
$$\begin{aligned}\text{(2)}\quad \text{residual}^{p2}_{h1} = {} & 0.25 * f\_vo^{p2}_{h1} - 0.25 * (f\_oo^{h2}_{h1} * t\_vo^{p1}_{h2}) + 0.25 * (f\_vv^{p1}_{p2} * t\_vo^{p2}_{h1}) \\ & +0.125 * (c\_vm^{p1}_{q1} * (it\_1^{q1h2}_{q2q3} * (c\_mv^{q1}_{p1} * (c\_mv^{q1}_{p2} * t\_vvoo^{p2p1}_{h1h2})))) \\ & -0.25 * ((f\_ov^{h1}_{p1} * t\_vo^{p1}_{h2}) * t\_vo^{p1}_{h2}) - 0.125 * ((t\_vo^{p2}_{h3} * v\_oovv^{h1h2}_{p1p2}) * t\_vvoo^{p2p1}_{h2h3}) \\ & -0.125 * ((t\_vvoo^{p1p2}_{h3h2} * v\_oovv^{h3h1}_{p1p2}) * t\_vo^{p1}_{h2}) + 0.25 * (t\_vvoo^{p2p1}_{h2h1} * it\_6^{h2}_{p2}) \\ & -0.25 * ((it\_6^{h1}_{p1} * t\_vo^{p1}_{h2}) * t\_vo^{p1}_{h2}) - 0.25 * (c\_vm^{p1}_{q1} * (c\_mo^{q2}_{h1} * it\_5^{q1}_{q2})) \\ & -0.25 * (c\_vm^{p1}_{q1} * (it\_3^{q2}_{h1} * it\_5^{q1}_{q2})) + 0.125 * (it\_4^{h2h3}_{h1p2} * t\_vvoo^{p2p1}_{h3h2}) \\ & -0.25 * ((it\_4^{h3h1}_{h2p1} * t\_vo^{p1}_{h3}) * t\_vo^{p1}_{h2}) + 0.25 * (t\_vvoo^{p2p1}_{h2h1} * f\_ov^{h2}_{p2})\end{aligned}$$

Fig. 4. The optimized formulation of CCSD T1. For compactness, summations are implicit wherever the same index appears twice in a term.

structure methods: The Tensor Contraction Engine. *Molecular Physics*, 104(2):211–218, 20 January 2006.

3. G. Baumgartner, A. Auer, D. Bernholdt, A. Bibireata, V. Choppella, D. Cociorva, X. Gao, R. Harrison, S. Hirata, S. Krishnamoorthy, S. Krishnan, C. Lam, Q. Lu, M. Nooijen, R. Pitzer, J. Ramanujam, P. Sadayappan, and A. Sibiryakov. Synthesis of high-performance parallel programs for a class of ab initio quantum chemistry models. *Proceedings of the IEEE*, 93(2):276–292, February 2005.
4. A. Hartono, A. Sibiryakov, M. Nooijen, G. Baumgartner, D. Bernholdt, S. Hirata, C. Lam, R. Pitzer, J. Ramanujam, and P. Sadayappan. Automated operation minimization of tensor contraction expressions in electronic structure calculations. In *Proc. ICCS 2005 5th International Conference*, volume 3514 of *Lecture Notes in Computer Science*, pages 155–164. Springer, May 2005.
5. H. Koch, O. Christiansen, R. Kobayashi, P. Jørgensen, and T. Helgaker. A direct atomic orbital driven implementation of the coupled cluster singles and doubles (CCSD) model. *Chem. Phys. Lett.*, 228:233, 1994.
6. C. Lam, P. Sadayappan, and R. Wenger. On optimizing a class of multi-dimensional loops with reductions for parallel execution. *Parallel Processing Letters*, 7(2):157–168, 1997.
7. G. Scuseria, C. Janssen, and H. Schaefer. An efficient reformulation of the closed-shell coupled cluster single and double excitation (CCSD) equations. *The Journal of Chemical Physics*, 89(12):7382–7387, 1988.
8. A. Sibiryakov. Operation Optimization of Tensor Contraction Expressions. Master's thesis, The Ohio State University, Columbus, OH, August 2004.
9. J. Stanton, J. Gauss, J. Watts, and R. Bartlett. A direct product decomposition approach for symmetry exploitation in many-body methods. I. Energy calculations. *The Journal of Chemical Physics*, 94(6):4334–4345, 1991.